INJECTING LINGUISTIC FEATURES AND PRE-TRAINED LANGUAGE MODEL IN NEURAL SEMANTIC PARSING

NGUYEN MINH PHUONG

Japan Advanced Institute of Science and Technology

Doctoral Dissertation

INJECTING LINGUISTIC FEATURES AND PRE-TRAINED LANGUAGE MODEL IN NEURAL SEMANTIC PARSING

NGUYEN MINH PHUONG

Supervisor : NGUYEN Le Minh

Graduate School of Advanced Science and Technology
Japan Advanced Institute of Science and Technology
Information Science

September, 2022

Abstract

Nowadays, in the quick development of science and technology, human life is getting more and more modern convenient with machines. Especially, computers and the internet are the major factors that help people connect together by storing, sharing, and searching for knowledge information in any domain. However, most of this information is written in unstructured data using human natural language, which is hard for logical searching as well as retrieval the meaningful information given complex questions. Therefore, this study aims to investigate the Semantic Parsing task in Natural Language Processing (NLP) that map a natural language sentence to machine-understandable information representation. In the developing history of this task, there are many kinds of semantic representations have been introduced and developed such as logical form, semantic frame, semantic graph, etc.

In this thesis, we introduce the effective methods using the neural network to solve the Semantic Parsing task. We focus on two kinds of widely used logic representations, logical form, and semantic frame as well as the issues of these semantic schemes. To this end, we propose the potential approaches to deal with the challenges of Semantic Parsing task, and present powerful methods for this tasks in the legal domain.

The first challenge we targeted is the local context integration in Semantic Parser. Inspired by grammar-based methods, the semantic representation of a sentence is the combination of sub-meaning representation generated by phrases in a sentence. Besides, the current state-of-the-art models using Transformer adapted from Neural Machine Translation task do not have components modeling phrase information. Therefore, we propose the Phrase Transformer - a new architecture incorporating representation of phrase via n-gram chunking into Self-Attention mechanism of the original Transformer. Our experimental results show that the proposed model works effectively and beat the original Transformer by utilizing local context features better.

The second issue we explored is the class imbalance in logical representation using semantic frames. The significant difference between the majority and minority classes causes the semantic parsing model confused in minority classes recognition. The observations on well-known datasets show that this problem is highly critical, special among Slot classes. To deal with this problem, we propose the Classify Anonymous Entities (CAE) mechanism by using multitask joint-learning to split the conventional Slot Filling task into two sub-task: detect anonymous entity by sequence tagging and classify recognized anonymous entities tasks.

Finally, we focus on constructing the semantic parser in the legal domain. The main challenges relate to the length and content of legal documents containing complex constraints about the conditions of articles. Besides, the limited annotated semantic parsing data also is a difficulty in this domain. Based on the DAPRECO Knowledge Base (KB), we firstly re-construct the GDPR (General Data Protection Regulation) Semantic Parsing dataset mapping a GDPR article points into its expression in DAPRECO KB. We also implement a Semantic Parser on this data and propose two mechanisms: Sub-expression intersection and Predicate REtrieval & Sub-Expression Generation (PRESEG) to deal with the problems in the legal domain.

To summarize, our study is centered on dealing with fundamental problems to model Semantic Parser using a deep learning approach and adapting to the legal domain. The experimental results and detailed analysis proved the effectiveness of the proposed methods as well as the potential for domain adaptation. Despite the experiments being conducted on limited kinds of semantic schemes, the proposed models and solution ideas have the potential to be widely applied to other types of semantic representations or to various tasks in NLP in futher research.

Keywords: Semantic Parsing, Phrase Transformer, Neural Machine Translation, Class Imbalance, Spoken Language Understanding, Legal Semantic Parsing.

Acknowledgments

Firstly, I would like to express my best sincerest gratitude to my supervisor during period of my research, Professor Nguyen Le Minh of Japan Advanced Institute of Science and Technology (JAIST). He inspired me in finding the appropriate research direction as well as taught me how to deal with problems in study.

I am deeply to thank the committee members: my second supervisor Associate Professor Kiyoaki Shirai, Associate Professor Inoue Naoya, Professor Satoshi Tojo, and Professor Ken Satoh at the National Institute of Informatics for useful suggestions and comments on my study. Through discussions, they help me recognize the limited points of my research as well as provided useful suggestions for improving the thesis.

I would like to express my appreciation to the Ministry of Education, Culture, Sports, Science and Technology for granting me a Doctoral Research Fellow scholarship during my Ph.D. time. I also pass on my thanks to the "JAIST Research Grant for Students" for supporting me to attend and presenting my research at international conferences.

I also would love to thank the JAIST staff, Football, and Badminton clubs for creating a wonderful environment for both work and life. Besides, I would like to devote my sincere thanks and appreciation to all Nguyen's laboratory members. The time I study and lived at JAIST is a wonderful time in my research life.

Finally, I would like to express my sincere thanks to my parents who always supported me with love and great patience. Without their support, I might never complete this work.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

Currently, computers and the internet are one of the most important factors in human life. Data is digitized in most fields and professions of life. In the digital age, more and more human-generated text data is created by the time such as articles, blogs, advertisements, etc. These data contain human intellectual information in an unstructured form, which is hard for organization as well as retrieval of the meaningful information given complicated questions. Therefore, this study aims to investigate the Semantic Parsing task in Natural Language Processing (NLP) that builds a system to parse a natural language utterance to its structured (machine-understandable) representation. Besides, this task also plays a key role in the human-machine communication field [Woods, 1973, Herzig and Berant, 2019, Jia and Liang, 2016]. For examples, the virtual assistants or smart speakers (e.g. Google Home, Amazon Alexa) became popular rapidly in recent times [Herzig and Berant, 2019]. Therefore, we expect that this study can be widely applied not only in the research community but also in practical application to improve human life.

1.2 Background

Recently, the development of machine learning and deep learning is extremely fast, especially in the NLP field. In that context, semantic analysis tasks are also of great interest in the research community. In this part, we show an overview of Semantic Parsing task, the problems and directions of our research.

1.2.1 Semantic schema

As we mentioned above, Semantic Parsing is the task mapping from a natural sentence into its logical representation. In the developing history of this task, many kinds of semantic representations have been introduced and developed. The most prominent are schemas using the logical form [Zelle and Mooney, 1996], and semantic frame [Fillmore and Baker, 2001]. For the syntax of logic representation, the logical forms typically use Lambda (λ) calculus or Prolog syntax, semantic frames with Intent, and Slot information. Both kinds of semantic representation are able to present detailed information in the sentence. In Table 1.1, we show examples to compare the difference between kinds of Logic schemas.

Table 1.1: Examples of Semantic Parsing task on ATIS dataset. The abbreviations LF and SF refer to two kinds of logic presentation logical form using lambda calculus and semantic frame, respectively.

Natural sentence	**Schema**	**Logic representation**
which *delta* flights fly from *boston* to *philadelphia*	LF	`( lambda $0 e ( and ( flight $0 ) ( airline $0 dl :al ) ( from $0 boston :ci ) ( to $0 philadelphia :ci ) ) )`
	SF	**Intent**: `atis_flight` **Slots**: `airline_name:` delta `fromloc.city_name:` boston `toloc.city_name:` philadelphia
what is the distance from *boston airport* to *boston*	LF	`( lambda $0 e ( and ( miles_distant $0 ) ( to_city $0 boston :ci ) ( from_airport $0 bos :ap ) ) )`
	SF	**Intent**: `atis_quantity` **Slots**: `fromloc.airport_name:` boston airport `toloc.city_name:` boston

Logical form

Lambda (λ) calculus is a formal system in mathematical logic that is introduced by Church [1941], that is typically used for formalizing the meanings of programming languages. Although this kind of meaning representation uses a mathematical formal language, it is close to the human natural language and easy to understand. In the first example in Table 1.1, the user want to search the flights of "Delta" airline, which fly from "Boston" city to "Philadelphia" city. In the lambda expression of this example, the variable `$0` save information of the flights need to be return. The structure information is constructed by open "`(`" and close "`)`" bracket pairs (Figure 1.1). The sub-expressions `(flight $0)`, `(airline $0 dl :al)`, `(from $0 boston :ci)`, and `(to $0 philadelphia :ci)` defines the semantic types, and related conditions in user question of variable `$0`, respectively. Finally, the function `and` is used to combine all sub-expressions to get the full logic representation.

```
( lambda $0 e
        ( and
            ( flight $0 )
            ( airline $0 dl :al )
            ( from $0 boston :ci )
            ( to $0 philadelphia :ci ) ) )
```

Figure 1.1: The structure of logical form using lambda calculus syntax.

To parse a natural sentence to logical form using lambda calculus syntax, a semantic parser need to deal with challenges such as the structure of logical form [Dong and Lapata, 2016], retrieving the related functions or compositional operators [Wang et al., 2015], and the corresponding arguments (e.g. string constants *boston*, *philadelphia*) [Jia and Liang, 2016, Nguyen et al., 2019] given the input sentence.

Semantic frame

The Semantic Frame is a type of knowledge representation introduced by Fillmore and Baker [2001], which construct the meaning of a sentence via defined frames and slots information. The main idea of this approach is to create a bank of semantic frames

(e.g. FrameNet [Ellsworth et al., 2021], PropBank [Palmer et al., 2005]), with each frame being an unambiguous meaning containing related slots information as arguments. In the task-oriented dialog system, this kind of semantic schema is regularly used because of its simplicity [Tur and De Mori, 2011, Chen et al., 2019, Qin et al., 2021b]. In the first example in Table 1.1, the semantic frame information is considered as intent value `atis_flight`, which contains slots `airline_name`, `fromloc.city_name`, and `toloc.city_name` are filled by string constant extracted from the input sentence.

In this approach, a semantic parser needs to deal with two main challenges: semantic frame, and slots (or entity) recognition [Tur and De Mori, 2011]. Compared with the approach using logical form, the output of the parser is well formed without syntax errors.

1.2.2 Methods

In this section, we present several methods that currently exist for the Semantic Parsing task grouped by Semantic Schemas. With the semantic representation using the logical form, one of the first methods is grammar-based and rule-driven semantic interpretation procedures [Waltz and Goodman, 1977]. Then, the grammar-based methods incorporating with probability models (e.g. Probabilistic Categorial Grammar) are introduced and developed [Zettlemoyer and Collins, 2005, 2007]. In recent years, with the great development in parallel computing using GPU, the approaches using neural networks have been applied with outstanding efficiency [Dong and Lapata, 2016, Jia and Liang, 2016, Dong and Lapata, 2018, Cao et al., 2019]. With the semantic representation using the semantic frame, the neural network approach also achieved the SOTA results such as using Recurrent Neural Network (RNN)-based [Ravuri and Stolcke, 2015, Wang et al., 2018], Self-Attention-based with Transformer [Vaswani et al., 2017b] model with pre-trained language model [Chen et al., 2019, Castellucci et al., 2019b, Qin et al., 2020, 2021a]. Based on the results achieved recently, in this study, we focus on the neural network methods and deal with some issues in this approach. For more understanding, we describe the basic architectures for tasks: *Sequence Generation* and *Sequence Labeling and Classification*.

Sequence Generation

Sequence-to-sequence. This model architecture is firstly proposed by Sutskever et al. [2014] for machine translation task and achieves a drastic improvement when incorporating the attention mechanism [Bahdanau et al., 2015, Luong et al., 2015]. This architecture contains two main components: Encoder and Decoder (Figure 1.2). The encoder uses

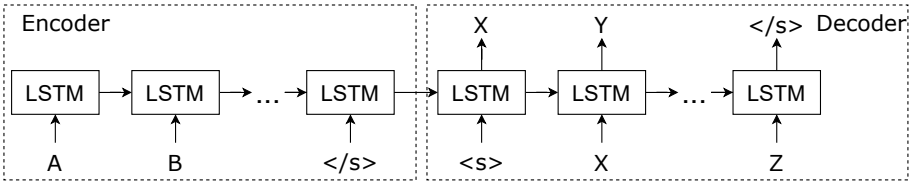


Figure 1.2: The architecture of basic Sequence-to-sequence model where special tokens <s>, </s> refer to the start and end of sentence; the the source and target words are denoted by [A, B, ..., </s>] and [<s>, X, Y, ..., </s>], respectively.

Long Short-Term Memory (LSTM) architecture [Hochreiter and Schmidhuber, 1997] to encode the source sentence as a compression vector, and the decoder also uses LSTM architecture to decode the encoded vector into the sentence in the target language.

To apply this architecture in the semantic parsing field, Dong and Lapata [2016] firstly promoted an effective solution by considering semantic parser as a neural translation model. In this way, the natural sentence and its logical form are treated as a source and target sentence, respectively. Finally, a trained model can learn the alignments among words in the natural sentence with tokens in the logical form.

Transformer. Recently, a new powerful model, Transformer, introduced by Vaswani et al. [2017b] got impressive performance in machine translation tasks by using the self-attention mechanism. Similar to the previous architecture, this model also contains two components Encoder and Decoder, separately. Compared with the model Sequence-to-sequence using LSTM, this architecture is based on the attention score between pairs of words to compute the dependencies between them. Therefore, it can overcome the vanishing gradient problem with the long sentence. Besides, this architecture is proven to be effective in transferring knowledge with pre-trained language models, especially on

machine reading comprehension tasks [Devlin et al., 2019b]. Therefore, our work focus on improving Transformer architecture for the Semantic Parsing task.

Sequence Labeling and Classification

As we mentioned above, on the task-oriented dialog system with logic representation using the semantic frame, the semantic parsing process can be addressed by joint Intent prediction, and Slot recognition sub-tasks [Tur and De Mori, 2011]. To solve both two sub-tasks, the architecture LSTM can be used [Ravuri and Stolcke, 2015, Wang et al., 2018]. With the Intent prediction task, the sentence representation vector is the last hidden state of sequence input, which is feed-forwarded to an output layer (e.g. softmax layer) to get the probabilities in each intent class. With the Slot recognition task, the hidden state of each word is used to get the probabilities in each slot class. Slot classes are typically followed the BIO schema similar to the other sequence labeling tasks in NLP (e.g. Named Entity Recognition - NER). Finally, the trained model can learn the relation between input words and their labels as well as the intent of whole sentence.

Recently, a new powerful pre-trained language model BERT is firstly introduced by Devlin et al. [2019a] achieved amazing results in Machine Reading Comprehension tasks with only small number of fine-tuning epochs. In semantic parsing tasks, this model also shows the strong improvement [Chen et al., 2019, Castellucci et al., 2019b] (Figure 1.3). To deal with the Sequence Labeling sub-task, the hidden states of each word also are

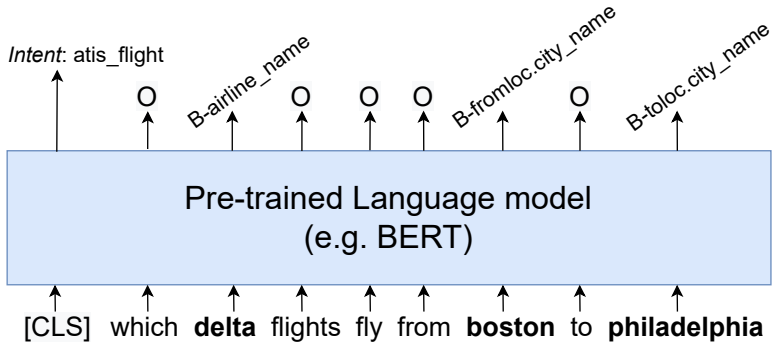


Figure 1.3: The architecture for joint of Intent classification and Slot recognition tasks.

used for calculating Slot label probabilities similar to LSTM architecture. In the Intent

detection sub-task, the hidden state of special token *[CLS]* is used for intent prediction. Compared with the model using LSTM, BERT model has more advantages because it is pre-trained on large-scale text data, which contains a huge human knowledge insight.

1.3 Research Direction and Contribution

In the quick development of machine learning and NLP recently, in the semantic parsing area, there are many kinds of semantic schema, and the model is introduced and investigated. In this research, our target is to obtain efficient methods for semantic parsing tasks in NLP related to two semantic schemes: logical form and semantic frame. We explore the basic problems this tasks: a fundamental problem of local context modeling, the imbalance classes related to semantic parser using sequence labeling approach, and domain-specific problems related to constructing a semantic parser in the legal domain. Based on the SOTA results recently, we focus on applying the Transformer architecture [Vaswani et al., 2017b] and pre-trained language model (e.g. BERT [Devlin et al., 2019a]) to this task. To this end, our study improved the semantic parsing system in three problems: local context modeling, diminishing class imbalance in Spoken Language Understanding, and dealing with complex constraints in parsing legal domain.

Local Context Integration. Inspired by grammar-based methods, the semantic representation of a sentence is the combination of sub-meaning representation generated by phrases in a sentence [Zettlemoyer and Collins, 2005]. Besides, the current SOTA models adapted from Neural Machine Translation task treat the sentence input as a word sequence that do not have components modeling phrase information. For example, a sentence "which delta flights fly *from Boston to Philadelphia*", the phrase "from Boston to Philadelphia" has a different meaning with "from Philadelphia to Boston". However, based on our error analysis, the Transformer model is confusing between them. Therefore, we propose the Phrase Transformer - a new architecture incorporating representation of phrase via n-gram chunking into Self-Attention mechanism of the original Transformer. In

detail, LSTM architecture is used for phrase modeling, reinforcing the linguistic ordering features of words in a phrase.

The experimental results show that our PhraseTransformer beat the original Transformer by utilizing local context features better. Besides, the extensive experiments on the Machine Translation task also show a solid improvement proving the generalize of our proposed model.

Class Imbalance in Spoken Language Understanding. As we mentioned, the semantic parsing task in task-oriented dialog systems (or Spoken Language Understanding) is addressed by Slot Filling and Intent Detection sub-tasks. The number of classes of the Slot Filling sub-task is typically large and unbalancing among classes. Especially, the imbalance is more critical between minority classes and negative class (the *outside entity* class - label `O`). This problem causes the semantic parsing model confused in minority class recognition. To deal with this problem, we propose the Classify Anonymous Entities (CAE) mechanism by splitting the conventional Slot Filling task into two sub-tasks, detecting anonymous entities by sequence tagging and classifying recognized anonymous entities, and using multitask joint-learning to train end-to-end model.

According to the experimental results, our proposed mechanism enhances semantic parsing performance compared with the conventional model, notably in the Slot Filling sub-task. Besides, we also present the effective way of integrating local context into the pre-trained language model and its contribution to this task. In addition, our experiments on the NER task also show the improvement that is proof of the applicability of our CAE mechanism to other tasks using sequence labeling.

Semantic Parsing in the Legal Domain. In the legal domain, the main challenges relate to the length and content of legal documents containing complex constraints about the conditions of articles. Besides, the limited annotated semantic parsing data also is a difficulty in this domain. Therefore, we firstly re-construct the GDPR (General Data Protection Regulation) Semantic Parsing dataset based on the DAPRECO Knowledge

Base (KB), which contains pairs of a GDPR article points in its expression in DAPRECO KB. We also implement a Semantic Parser on this data and propose two mechanisms: Sub-expression intersection and Predicate REtrieval & Sub-Expression Generation (PRESEG) to deal with the problems in the legal domain. Based on the experimental results, our proposed mechanisms show better performance when compared with the baseline model. Furthermore, we also conduct experiments integrating local context into the semantic analysis model in this domain and show improved results.

1.4 Dissertation Outline

Firstly, we investigate the effect of local context problems in the semantic parsing task with two kinds of semantic schema, logical form and semantic frame, which is usually represented by phrases in natural sentences [Zettlemoyer and Collins, 2005, 2007, Jia and Liang, 2016]. Local context is a fundamental problem, which exists in almost semantic analyses system such as semantic searching, Spoken Language Understanding (SLU) of Virtual Assistant, etc.

In addition, we focus on the problem of data imbalance between classes of a semantic parsing system representing meaning via semantic frames as a with intent and slots information. Compared to the first problem, the imbalance problem is occurring to a narrower extent, in systems using sequence labeling mechanisms. This problem typically does not exist in semantic parsing systems using logical form because these systems do not require label all words in a natural sentence. However, with the rapid development of virtual assistant systems today [Herzig and Berant, 2019], the imbalance problem in SLU has high applicability in practical applications as well as the other tasks such as Named Entity Recognition, POS tag, etc.

Finally, we apply SOTA methods in semantic parsing tasks for a particular Legal domain. In this challenge, we based on DAPRECO KB [Robaldo et al., 2020] to re-construct the semantic representation of the law articles and build a semantic parser for the GDPR dataset. Similar to logical form, the logic representation in this data is the combination

of formulae containing many sub-conditions or triples. Besides local context problems, we also focus on solving the particular problems related to the legal domain: the long document, and complex logic representation. To this end, our proposed solutions can be applied to many different kinds of legal data, or the other domain containing complicated in a long input sentence.

We have introduced the abstract as well as presented the research direction of our work in this Chapter. In the remainder of this thesis, we provide the detail of the experiments and our proposed model architecture following (Figure 1.4):

- Chapter 2 describes the detail of PhraseTransformer architecture improving sentence meaning representation by injecting phrase features as local context information. The experiments and results on three public Semantic Parsing datasets (Geo, Atis, and MSParS) along with three Machine Translation well-known datasets (IWSLT14 German-English, IWSLT15 English-Vietnamese, and WMT 2014 English-German) are illustrated and analyzed.

- Chapter 3 interprets our CAE mechanism for diminishing class imbalance in the Slot Filling of the SLU task. The conducted experiments on prominent SLU datasets Snips ATIS and ATIS Vietnamese version are presented.

- Chapter 4 reports our empirical evaluation of constructing a GDPR semantic parsing dataset along with a parsing model adapted to the legal domain. Besides, the error analysis of existing issues as well as the further solutions are discussed.

- Finally, in Chapter 5 we conclude our research with a summary of our findings throughout this dissertation and the future directions based on this work.

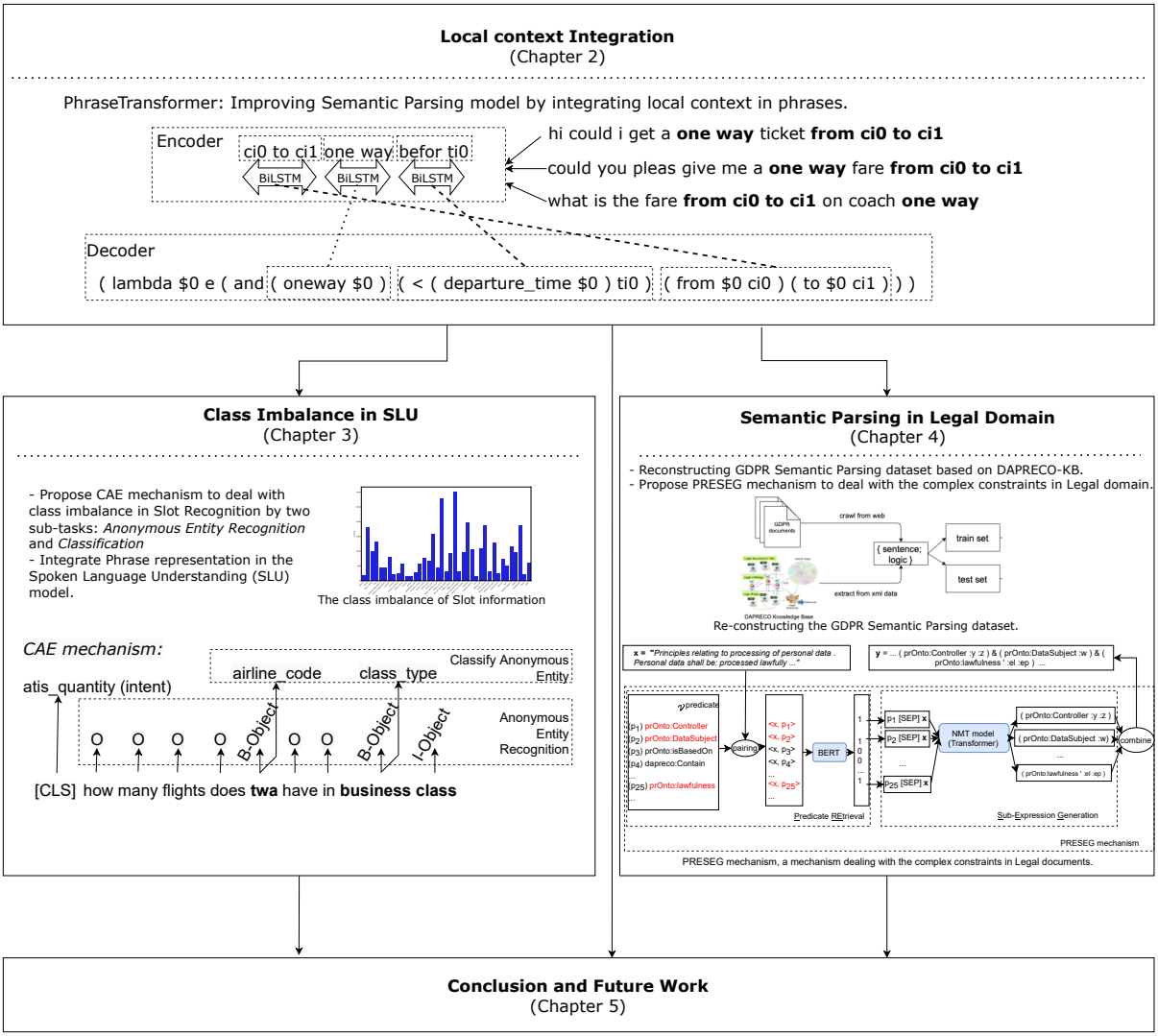


Figure 1.4: The overall of our work and the relations among problems.

Chapter 2

Local Context Integration

Semantic parsing is a challenging task in NLP that attracts the attention of many researchers for a long. Recently, approaches using Neural Machine Translation (NMT) have achieved many promising results, especially Transformer, because of the ability to learn long-range word dependencies. However, the typical drawback of adapting the vanilla Transformer to semantic parsing is that it does not consider the phrase in expressing the information of sentences while phrases play an important role in constructing the sentence meaning. Therefore, we propose an architecture, PhraseTransformer, that is capable of a more detailed meaning representation by learning the phrase dependencies in the sentence. The main idea is to incorporate Long Short-Term Memory into the Self-Attention mechanism of the original Transformer to capture the local context of a word. Experimental results show that our proposed model captures the detailed meaning better than the original Transformer, and raises the model local context-awareness. Besides, the proposed model achieves strong competitive performance on Geo, and MSParS datasets, and leads to SOTA performance on the Atis dataset in methods using neural networks. In addition, to prove the generalization of our proposed model, we also conduct extensive experiments on three translation datasets IWLST14 German-English, IWSLT15 Vietnamese-English, WMT14 English-German, and show the solid improvement.

2.1 Introduction

Semantic parsing is an important task that can be applied in many applications such as Question Answering and searching systems using natural language [Woods, 1973, Waltz and Goodman, 1977]. For example, the sentence *"which state borders hawaii"* can be represented as a logical form (LF) using λ-calculus syntax *"(lambda $0 e (and (state:t $0) (next_to:t $0 hawaii)))"*. There are various strategies to address the semantic parsing task such as constructing handcraft-rules [Woods, 1973, Waltz and Goodman, 1977, Hendrix et al., 1978], using grammar-based (e.g. Combinatory Categorial Grammar - CCG) [Zettlemoyer and Collins, 2005, 2007, Kwiatkowski et al., 2011], adapting statistical machine translation method [Wong and Mooney, 2006, 2007], and Neural Machine Translation (NMT) [Dong and Lapata, 2016, Jia and Liang, 2016, Cao et al., 2019].

Among them, the grammar-based approaches are fundamental and effective to show how the logical form is constructed from sub-parts of a sentence. The major factor of this method is based on the alignments of sub-parts (lexicons or phrases) between a natural sentence and corresponding logical form and to learn how best to combine these sub-parts. In particular, the phrase *"borders hawaii"* is aligned to *"(next_to:t $0 hawaii)"* in LF and the word *"borders"* plays a role as a local context of *"hawaii"* that makes the meaning of this phrase more comprehensive than every single word. Conversely, the methods using Neural Machine Translation digest the sentence via an encoder into a context vector which is decoded into LF. Despite the lack of linguistic relationship in local context of sub-parts, the current SOTA models comes from NMT-based models such as Sequence-to-Sequence (Seq2seq) using Long Short-Term Memory (LSTM) [Dong and Lapata, 2018, Cao et al., 2019] on Geo, Atis and Transformer [Ge et al., 2019] on MSParS. The strength of NMT-based methods depends on the automatic text understanding architecture without any handcrafted features. However, recent NMT-based approaches in Semantic Parsing give a little attention to linguistic constraints and relationship in the typical characteristics of natural languages. Therefore, it is highly potential to improve these approaches by incorporating the local context of phrases whose effectiveness and necessity is proved in

many previous works of grammar-based approaches.

Inspired by grammar-based methods, the semantic representation of a sentence is the combination of sub-meaning representation generated by phrases in a sentence. However, Transformer architecture only learns the dependencies between single words without considering the local context by the phrase. Therefore, we propose a new architecture named PhraseTransformer that focuses on learning and integrating the relations of phrases in a sentence into the Transformer-based architecture. The typical example of our proposed approach is presented in Figure 2.1. Together with word representation, our incorporation of phrase information is useful enough to learn local context and relationship and understand the global information via Transformer-based architectures. Although there were numerous works considering to utilize phrase representation on NMT [Yang et al., 2018, Nguyen et al., 2020, Xu et al., 2020b], our proposed approach is highly novel and effective in Semantic Parsing. Instead of using handcrafted-features as well as syntactic information, our approach takes advantage of LSTM-based and Transformer-based approaches into a completed system to automatically understand the sentence representation.

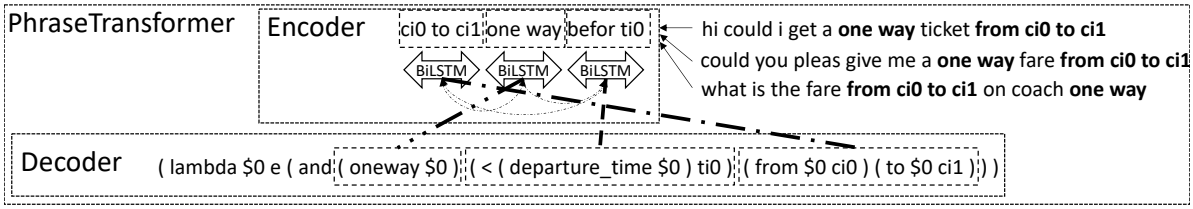


Figure 2.1: Pharse alignments in PhraseTransformer.

To this end, we particularly use the power of LSTM architecture to represent the local context meaning of n-gram phrases in the sentence. Then, we modify the Multi-head Attention [Vaswani et al., 2017b] in Transformer by applying the self-attention mechanism into phrases instead of single words. Based on the interaction of words and phrases, we propose two variants of our integration between local and lexical context in Transformer architecture. Both of them are proved the effectiveness in two important tasks of sequence learning including Semantic Parsing and Machine Translation. Even that, in Semantic Parsing task, our proposed models obtain the significant results on three

benchmark datasets including Geo, Atis, and MSParS. It highlights the importance of phrase information in current sentence representation approaches. In this work, our main contributions are:

- We propose a novel model, PhraseTransformer, incorporating phrase information as a local context into Transformer architecture that works effectively for Semantic Parsing and Machine Translation tasks. The phrase integration layer is lightweight and does not require any external information.
- We conduct experiments to confirm the awareness capacity of the model, as well as the contribution of phrase information in each task.
- Our work achieve competitive performance on Geo, MSParS datasets and new SOTA performance on Atis, in the methods using Neural Network. To the best of our knowledge, our work is the first to use phrase mechanism in using Sequence to sequence model to solve the Semantic Parsing task.
- The PhraseTransformer architecture can be adapted to many tasks using Transformer architecture such as Text Summarization and Text Classification.

2.2 Related Work

In **Semantic Parsing** task, recent works have shown that deep learning approaches achieved potential results. Traditionally, these methods are often divided into four groups:

Decoder Customization. Dong and Lapata introduce the Seq2tree model modifying the decoding method to deal with the tree structure of the LF. On another aspect, some works [Dong and Lapata, 2018, Li et al., 2019] focuses on the design of intermediate representations for LF, and the decoding process is split into two steps: generate the template of LF and fill the low-level information into the template. Pursuing a different direction, we tackle the challenge of improving the understanding capacity of the model in comprehending input sentences because semantic parsers need to capture complicated

characteristics in the natural sentences before decoding them. Therefore, our work focuses on designing the Encoder architecture to improve the understanding capacity of the model.

Data Augmentation. There are numerous works that put the concentration on data augmentation to improve the performance of the semantic parsing model [Berant and Liang, 2014, Wang et al., 2015, Jia and Liang, 2016, Ziai, 2019, Herzig and Berant, 2019]. Jia and Liang propose hand-craft rules based on Synchronous Context-Free Grammar to recombine data. This step increases the size of the training data and grows the performance of the model. Similarly, Ziai proposes a method that automatically augments data based on the co-occurrence of words in the sentence. These works suggest that the complicated queries are constructed by a combination of many sub-conditions that is phrases. Therefore, modeling phrases for Semantic Parsing is a potential way to enhance the generalize of the model.

Weak Supervision. Some typical approaches use semi-supervised learning for semantic parsing task such as [Kočiský et al., 2016, Yin et al., 2018, Goldman et al., 2018, Cao et al., 2019, 2020]. These works are promising approaches for the data-hungry problem because of the ability to extract latent information such as unpaired logical forms. In our proposed model, we aim to construct the latent representation for phrases and learn these representations via the self-attention mechanism of the Transformer. We hypothesize that complicated sentences are constructed from various phrases, so learning to represent these phrases makes the model more generalizable.

Sketch Prediction and Slot Filling. There are many recent works that pay attention to the approach using sketch (or intent) prediction and slot filling to deal with the Semantic Parsing task [He et al., 2021, Tang et al., 2020, Xu et al., 2020a]. Besides, SQL parsing is also one of the Semantic Parsing task attracting many works Xu et al. [2017], Bogin et al. [2019], Wang et al. [2020], Yu et al. [2021], Xie et al. [2021]. The main challenge in the SQL parsing task is the generalization of database schema such as the alignment between natural text and column names, primary, foreigner keys of a table. Differently from that, in this work, we focus on the challenge of generalizing the sub-conditions and relations among them in a complicated query by phrase modeling mechanism.

In **Neural Machine Translation** task, the approach using phrase information or constituent tree is proved to be effective in many works [Wang et al., 2017, Wu et al., 2018, Wang et al., 2019, Hao et al., 2019, Nguyen et al., 2020]. The difference in our work are: (1) our model is capable of learning without any additional information (e.g. constituent tree), (2) in the training process, although we do not force the attention or limit the scope of the dependencies, our model is able to pay high attention to the important phrase automatically. Compare with Yang et al. [2018], the purpose of using local context information is similar but different in *localness modeling*: based on the distance, Yang et al. [2018] cast a Gaussian bias to change attention score while our method is simpler by incorporating multi different n-gram views as the various local contexts. Compare with Xu et al. [2020b], the authors used numerous parameters to learn the attention score of each token in phrase representation that makes the model size is larger 2 times than the original Transformer while our proposed architecture size is close to the original model.

2.3 Model Architecture

Our novel architecture is based on the Encoder-Decoder of Transformer [Vaswani et al., 2017b]. We define a new model named *PhraseTransformer* to improve the encoding quality of Transformer by enhancing the Encoder architecture while keeping the original Decoder. Besides, we proposed two architecture variants to construct *phrase* representation appropriate for characteristics of Semantic Parsing and Machine Translation tasks: PhraseTrans.$_{SepH}$ and PhraseTrans.$_{CrossH}$.

2.3.1 Background

The Transformer model contains two parts Encoder and Decoder. In the basic setting, both parts contain similar architecture that is based on the self-attention mechanism to understand and extract linguistic features of words in a sentence. Particularly, in Transformer Encoder architecture, Vaswani et al. [2017b] proposed a stack of N Identical

layers which consists of two sub-layers: Multi-Head Attention layer and Position-wise Feed-Forward layer.

Mathematically, Let $\boldsymbol{x} = [\boldsymbol{x}_1, ..., \boldsymbol{x}_{|S|}]$ be an input vector synthesized from the vector word embedding and positional encoding where $|S|$ is sentence length. In the Multi-Head Attention layer, Vaswani et al. use the Linear layer to get multi-views for the inputs. This layer processes the input vector ($\boldsymbol{x}$) and generates H distinct featured vectors $\{\boldsymbol{q}_i, \boldsymbol{k}_i, \boldsymbol{v}_i\}_{i=1}^{H}$ where H is the number of heads (Equation 2.1). These features are forwarded to Self-Attention layer using Scaled-Dot Product (Equation 2.2). After that, all heads are processed by Concatenation operator and Linear layers to compute the output of the Multi-Head layer.

$$\boldsymbol{q}_i, \boldsymbol{k}_i, \boldsymbol{v}_i = \boldsymbol{x}\boldsymbol{W}_i^q, \boldsymbol{x}\boldsymbol{W}_i^k, \boldsymbol{x}\boldsymbol{W}_i^v \tag{2.1}$$

$$\boldsymbol{head}_i = \text{Attention}(\boldsymbol{q}_i, \boldsymbol{k}_i, \boldsymbol{v}_i) \tag{2.2}$$

$$\boldsymbol{h}_{MulH} = [\boldsymbol{head}_1; ...; \boldsymbol{head}_H]\boldsymbol{W}^o \tag{2.3}$$

$$\boldsymbol{h}_{Norm} = \text{LayerNorm}(\boldsymbol{h}_{MulH} + \boldsymbol{x}) \tag{2.4}$$

$$\boldsymbol{h}_{out} = \text{LayerNorm}(\text{FeedForward}(\boldsymbol{h}_{Norm}) + \boldsymbol{h}_{Norm}) \tag{2.5}$$

where $\boldsymbol{W}$ is the parameters; Attention is Scaled Dot-Product Attention as follows:

$$\text{Attention}(\boldsymbol{q}_i, \boldsymbol{k}_i, \boldsymbol{v}_i) = \text{softmax}(\frac{\boldsymbol{q}_i \boldsymbol{k}_i^\intercal}{\sqrt{d_h}})\, \boldsymbol{v}_i \tag{2.6}$$

where d_h is dimensions per head, i is the identical index of head ($0 < i \leq H$), LayerNorm, FeedForward are the functions that are used similar to Vaswani et al. [2017b].

2.3.2 Proposed Architecture

In Transformer architecture, Encoder layer plays an important role to digest and extract the textual features of input sequence. With our powerful Encoder, it is essential enough to address the lack of phrase information in vanilla Transformer approaches. In our proposed architectures, we propose a modification of the MultiHead Attention layer in the Encoder

by integrating the phrase features. Particularly, we add a new layer, Phrase Integration, before Scaled Dot-Product Attention layer. It is effective to provide the locally contextual information of words and their relationship into original Encoder. The overview of our model is visualized in Figure 2.2. Based on our observation, we propose two variant of Phrase Integration to combine the phrase information and word representation as follows:

- **PhraseTrans.**$_{SepH}$ is designed to have a relation modeling among phrases. In this architecture, we apply different n-gram models to make various phrase representations in each separated head in the Multi-Head layer. After that, the phrase representation is forwarded into the Self-Attention layer to extract the relations between them.

- **PhraseTrans.**$_{CrossH}$ is designed to improve word representation by incorporating phrases characteristic. Compare with the above architecture, in this architecture, we aim to mitigate the lost information of single words when integrating phrase information, therefore, we only apply phrase mechanism query and key vectors to learn the attention scores and keep the original value vectors. Besides, with each n-gram model, we apply the phrase mechanism in all heads of the Multi-Head layer. After that, the word representation is concatenated by phrase representation and forwarded to the Self-Attention layer.

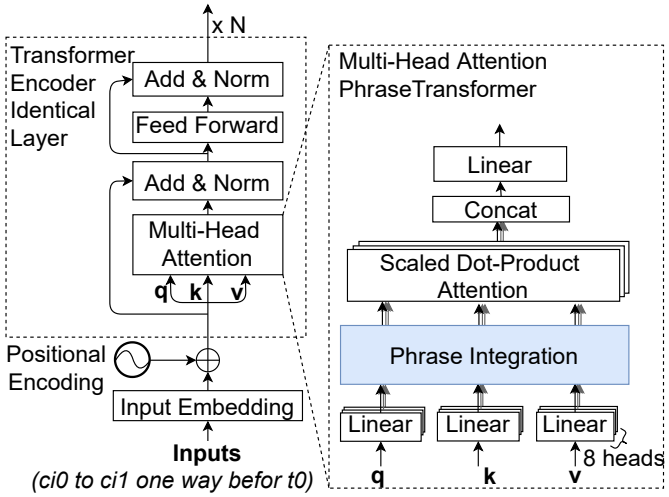


Figure 2.2: Overview of PhraseTransformer

Phrase Integration on Separated Head (PhraseTrans.$_{SepH}$) We replace the word vector representations in the MultiHead layer by its context representations that are combined by its n-gram words. More detail, after H heads are generated by Linear layer, we use n-gram model to split the sentence into grams and use Bidirectional LSTM [Hochreiter and Schmidhuber, 1997] to extract the local context information of these grams (Figure 2.3). Besides, we assume that the meaning phrases are usually composed

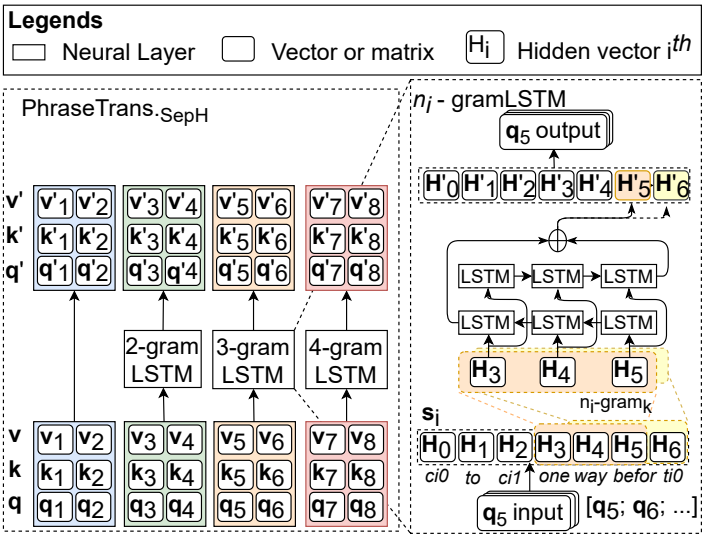


Figure 2.3: PhraseTransformer Encoder architecture using n_gram LSTM in MultiHead layer. In this case, n_gramLSTM layer is built with $\boldsymbol{n} = [0, 0, 2, 2, 3, 3, 4, 4]$, 2-gram, 3-gram, 4-gram models apply to every two heads from head 3 to head 8.

by difference lengths, therefore we use various n-gram models. To this end, the Phrase function is in Equation 2.7:

$$\text{Phrase}(\boldsymbol{s}_i, n_i) = \begin{cases} n_i\text{\_gramLSTM}(\boldsymbol{s}_i) & \text{if} \quad n_i \neq 0 \\ \boldsymbol{s}_i & \text{otherwise} \end{cases} \tag{2.7}$$

where $\boldsymbol{s}_i$ is a sequential hidden state of a sentence of head i $(0 < i \leq H)$ in Multi-Head layer; $\boldsymbol{n} \in \mathbb{N}^H$ is gram size vector for H heads; n_i is the gram size corresponding to head i; n_i_gramLSTM is a procedure that splits the sequential input into grams by n_i_gram

model, and applies Bidirectional LSTM for each gram k of $\boldsymbol{s}_i$:

$$n_i\text{-gramLSTM}(\boldsymbol{s}_i) = [n_i\text{-gramLSTM}_k(\boldsymbol{s}_i)] \tag{2.8}$$

where n_i-gramLSTM$_k$ is the Bidirectional LSTM computed by sum of forward and backward final hidden states:

$$n_i\text{-gram}_k(\boldsymbol{s}_i) = [\boldsymbol{H}_{k-n_i+1}, \boldsymbol{H}_{k-n_i+2}, ..., \boldsymbol{H}_k] \tag{2.9}$$

$$n_i\text{-gramLSTM}_k(\boldsymbol{s}_i) = \text{LSTM}_i^f(n_i\text{-gram}_k(\boldsymbol{s}_i)) + \text{LSTM}_i^b(n_i\text{-gram}_k(\boldsymbol{s}_i)) \tag{2.10}$$

where $\boldsymbol{H}_k$ is the hidden state corresponding to word index k in a sentence, n_i-gram$_k$ is the gram index k that is a list of n_i continuous hidden states, n_i-gramLSTM$_k(\boldsymbol{s}_i)$ is the vector to capture local context of the gram index k. Besides, for parallel computing, we used the left padding zero mechanisms to get the same shape between input and output when applying different n-gram models. Finally, the query ($\boldsymbol{q}_i$), key ($\boldsymbol{k}_i$), value ($\boldsymbol{v}_i$) matrixes (Equation 2.2) are replaced by *Phrase* function:

$$\boldsymbol{q}'_i, \boldsymbol{k}'_i, \boldsymbol{v}'_i = \text{Phrase}(\boldsymbol{q}_i, n_i), \text{Phrase}(\boldsymbol{k}_i, n_i), \text{Phrase}(\boldsymbol{v}_i, n_i) \tag{2.11}$$

$$\boldsymbol{head}_i = \text{Attention}(\boldsymbol{q}'_i, \boldsymbol{k}'_i, \boldsymbol{v}'_i) \tag{2.12}$$

Phrase Integration Cross Heads (PhraseTrans.$_{CrossH}$) Compared with the previous PhraseTrans.$_{SepH}$, in this architecture, we construct phrase representation across all heads with each n-gram model. After that, we replace the original query, key vectors in each head by the concatenation of single words vectors and phrase vectors and forward them to the Self-Attention layer (Figure 2.4).

We use a new hyper-parameters $\mathbf{m} = \{m_t \mid m_t \in \mathbb{N}\}$ to store the set of gram sizes applied to all heads in the Multi-Head layer. Although this hyperarameters open a unlimited searching space, our experiments show that $\mathbf{m} \in \{\{2,3\},\{3\},\{4\}\}$ help model get the best performance on many NMT datasets. Let zip be the concatenation function in

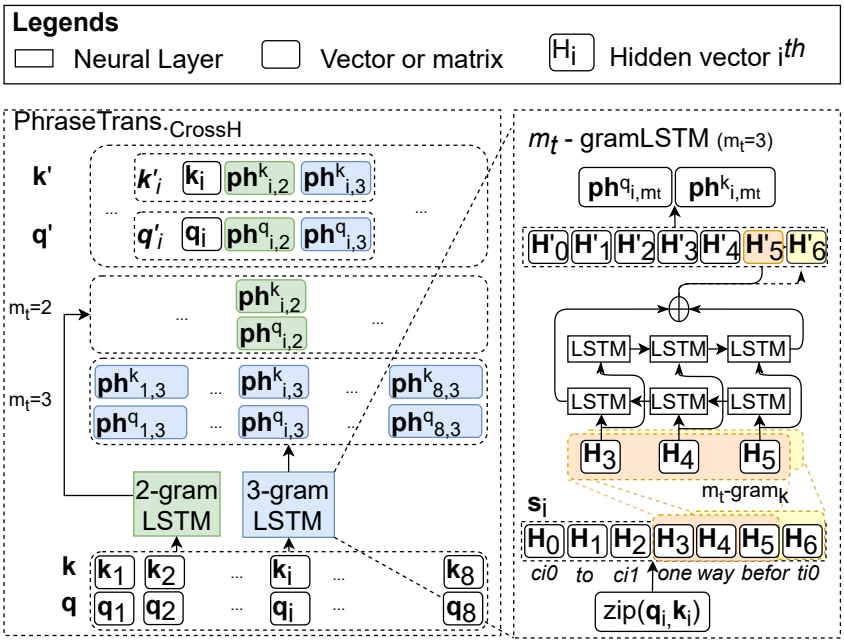


Figure 2.4: PhraseTrans.$_{CrossH}$ architecture using n_gram LSTM in MultiHead layer. In this case, the phrase representations are built with $\boldsymbol{m} = \{2, 3\}$, 2-gram, 3-gram models apply to all 8 heads.

the second dimension of a matrix (Equation 2.13), and unzip be the reverse function of zip. The query and key matrices in Equation 2.2 are replaced by a zip of word and phrase vector representations:

$$\text{zip}(A, B) = [A^\intercal; B^\intercal]^\intercal \tag{2.13}$$

$$\boldsymbol{q}'_i,\ \boldsymbol{k}'_i = \text{zip}(\boldsymbol{q}_i, \boldsymbol{ph}^q_i),\ \text{zip}(\boldsymbol{k}_i, \boldsymbol{ph}^k_i) \tag{2.14}$$

$$\boldsymbol{head}_i = \text{Attention}(\boldsymbol{q}'_i, \boldsymbol{k}'_i, \boldsymbol{v}_i) \tag{2.15}$$

where $\boldsymbol{q}'_i, \boldsymbol{k}'_i \in \mathbb{R}^{|S| \times (|\boldsymbol{m}|+1)*d_h}$ are the new query, key vectors fused phrase features, respectively; $|S|$ is a sentence length; $|\boldsymbol{m}|$ is the number of applied n-gam models. Here, we remind that in the original Transformer, the query, key vectors in the head i^{th} ($\boldsymbol{q}_i, \boldsymbol{k}_i \in \mathbb{R}^{|S| \times d_h}$) actually are the weighted hidden states of sentence generated by Equation 2.1. Compared with the PhraseTrans.$_{SepH}$ architecture each head use one kind of n-gram model, while in this architecture, we aim to inject many kinds of n-gram model into each head as a linguistic features for key, query vectors. For mathematically, phrase representation vectors for all gram sizes of query and key in the head i^{th} in MultiHead

layers ($\boldsymbol{ph}_i^q, \boldsymbol{ph}_i^k \in \mathbb{R}^{|S| \times |\boldsymbol{m}|^* d_h}$) are computed:

$$\boldsymbol{ph}_{i,m_t}^k, \boldsymbol{ph}_{i,m_t}^q = \text{unzip}(\text{ Phrase}(\text{zip}(\boldsymbol{q}_i, \boldsymbol{k}_i),\ m_t)\ ) \tag{2.16}$$

$$\boldsymbol{ph}_i^q = \text{zip}(\{\boldsymbol{ph}_{i,m_t}^q \mid m_t \in \boldsymbol{m}\}) \tag{2.17}$$

$$\boldsymbol{ph}_i^k = \text{zip}(\{\boldsymbol{ph}_{i,m_t}^k \mid m_t \in \boldsymbol{m}\}) \tag{2.18}$$

where the *Phrase* procedure of each gram size value (m_t) is computed similar to the PhraseTrans.$_{SepH}$ architecture. Beside, we use zip function on the query and key vectors of each head before forwarding it into LSTM architecture to utilize the relation between query and key vectors. The function zip($\boldsymbol{q}_i, \boldsymbol{k}_i$) generates a aggregating matrix in $\mathbb{R}^{|S| \times 2^* d_h}$ space, that is input of *Phrase* procedure. To this end, in this architecture, with each gram size value, we apply the phrase mechanism for all heads in the Multi-Head layer, and we can compute phrase vector representations of all heads by parallel processing in constant time.

Training method The training objective is to maximize the Log-Likelihood function of the probabilities to generate the LF (y) given a sentence (x) from annotated dataset ($\mathcal{D}$):

$$\text{maximize} : \sum_{<x,y> \in \mathcal{D}} \log\ p_\theta\left(y \mid x\right) \tag{2.19}$$

2.4 Experiments

The purpose of experiments is to compare the performance of PhraseTransformer and its variants models with the original Transformer on both Semantic Parsing and NMT tasks. Besides, we explore the awareness of the phrase alignment between a sentence and generated LF by PhraseTransformer.

2.4.1 Datatsets

We conduct the experiments on three datasets Geo [Zelle and Mooney, 1996], Atis [Dahl et al., 1994], MSParS [Duan, 2019] for Semantic Parsing task and three datasets IWLST14[1] (German ↔ English), IWLST15[2] (Vietnamese ↔ English), WMT14[3] (English → German) for NMT task. Table 2.1 shows the statistics of these datasets. Geo and Atis datasets are small in size but more complicated in nested conditions than the MSParS dataset. The average length of LFs on Atis dataset (28.4) is about twice longer than that on MSParS dataset (14.7). With both Atis and Geo, we used the version preprocessed by Dong and Lapata [2016] by replacing all entities by numbered markers (e.g. *"new york"* → *"s0"*). The original MSParS dataset have large vocabulary (around 40k) because it consists of various entities name in the open domain. Therefore, we preprocess this dataset similarly to Ge et al. [2019] by replacing character "_" by "<space>_<space>" and using the byte-pairs-encoding (BPE) [Sennrich et al., 2016] to deal with rare-word problem. For the NMT task, we preprocess all datasets with the Moses toolkit[4] for tokenizer, apply the BPE method and share vocabulary between source and target sides (Table 2.1).

Table 2.1: Statistics information of all datasets. Vocabulary size and average length of source (Src) and target (Tgt) side are computed on training set.

Datatset	**#examples**			**#vocab**		**Average length**	
	Train	Dev	Test	Src	Tgt	Src	Tgt
Geo	600	0	280	433	51	10.6	18.7
Atis	3434	491	448	120	166	7.3	28.4
MSParS	63K	9K	9K	5K	6K	12.8	23.9
IWSLT14 de↔en	160K	7K	7K	10K	10K	24.2	23.6
IWSLT15 vi↔en	133K	1.5K	1.3K	10K	10K	20.3	24.8
WMT14 en→de	4M	40K	3K	41K	41K	29.1	28.4

[1]http://workshop2014.iwslt.org/
[2]https://workshop2015.iwslt.org/
[3]http://www.statmt.org/wmt14/
[4]https://github.com/moses-smt/mosesdecoder

2.4.2 Evaluation Metric

On all datasets of the Semantic Parsing task, we report sentence-level accuracy by using logic matching (LM) that developed by Dong and Lapata [2018]. Base on the parsed structure of output logic form, LM metric is able to measure the variant of expression. For example, the predicted LFs in different order of *and* logic: *and (oneway $0) (<(departure_time $0) ti0)* is equal to *and (<(departure_time $0) ti0) (oneway $0)* . In the NMT task, we evaluated performance by averaging 5 latest checkpoints and compute BLEU score via SacreBleu [Post, 2018] on WMT 2014 datasets[5] and use multi-bleu script[6] on IWSLT 2014, 2015 datasets for comparable with previous published results. Besides, we also conducted significant test following Koehn [2004].

2.4.3 Settings

In training processes of all datasets (except Geo dataset), to prevent overfitting, we use the *early stopping* conditioned on metric accuracy on dev set. Because Transformer is quite sensitive in hyperparameters, we keep most hyperparameters the same as Transformer-base model [Vaswani et al., 2017b] such as the number of layers $N = 6$ and number of heads in Multi-Head layer is $H = 8$; hidden size $d_{model} = 512$; dropout is selected in $\{0.1, 0, 3\}$; Adam optimizer with $\beta_1 = 0.9, \beta_2 = 0.998, \epsilon = 10^{-9}$. The weights of models are initialized with Xavier initialization [Glorot and Bengio, 2010]. The embedding vectors are shared among the source-side and target-side, between the input-to-embedding layer and output-to-softmax layer in Decoder. We also retain the learning rate decay method: $lr(step) = d_{model}^{-0.5} \cdot \min(step^{-0.5}, step \cdot warmup\_steps^{-1.5})$ where *step* is the current step number. The n-gram size for each head is selected in $\{0, 2, 3, 4\}$. The hyperparameter gram sizes are fine-tuned on the development set to get the best setting and re-evaluate on the test set. With the Geo dataset, because this dataset does not contain the development set, we adapted the best setting of the Atis dataset to this dataset, and run five times in

[5] Our SacreBleu signature: `BLEU+case.mixed+lang.en-de+numrefs.1+smooth.exp +test.wmt14/full +tok.13a+version.1.5.1`

[6] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu.perl

different random seeds and report the average value of the last check point. The weights of Bidirectional LSTM layers in the heads using the same n-gram model (e.g. heads 3, 4 in Figure 2.3) are shared.

In **Semantic Parsing** task, the experimental dataset sizes are quite different, therefore we use three hyper-parameter sets[7]: **Geo**: $warmup\_step = 100$ learning rate init selected from $\{\mathbf{0.05}, 0.1\}$, $batch\_size = 128$ (the batch size using number of tokens), the maximum training step $max\_steps = 15000$; **Atis**: $warmup\_step = 100$; learning rate init selected from $\{\mathbf{0.1}, 0.2\}$, $batch\_size = 4096$, the maximum training step is 250000; **MSParS**: $warmup\_step = 8000$, $batch\_size = 8192$, $max\_steps = 250000$. On this dataset, we conducted preliminary experiments to check the number of BPE operations in 6K, 8K, 12K, 16K (Figure 2.5). Based on those results, we use the MSParS dataset preprocessed by BPE 6000 operations for all other experiments.

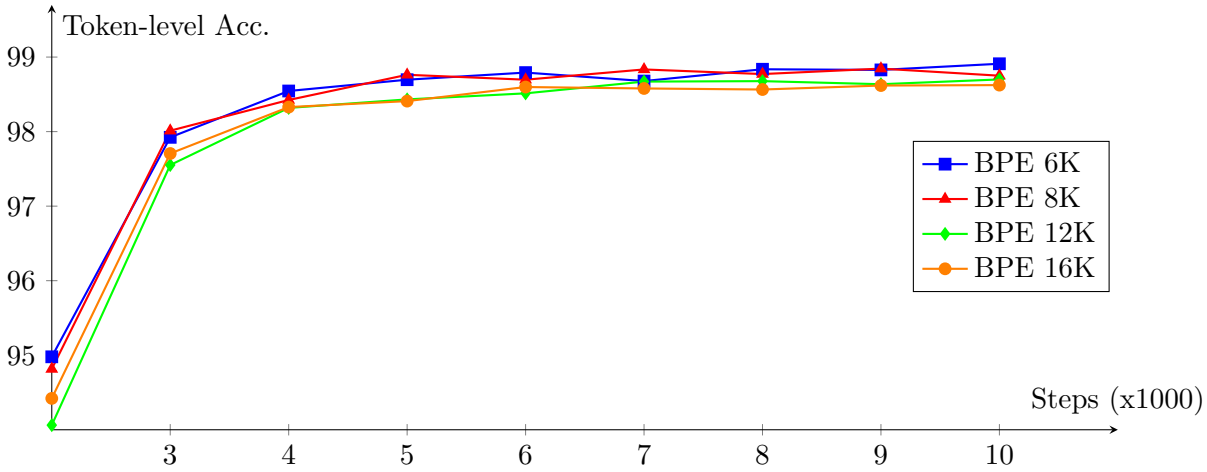


Figure 2.5: The impact of BPE preprocessing to performance of PhraseTransformer on MSParS dev set.

In **Machine Translation** task, we also use Transformer base setting. All datasets are pre-processed tokeinze with the standard Moses toolkit[8]. With **IWSLT14**, **IWSLT15** datasets: $warmup\_step = 4000$, $batch\_size = 4096$ tokens, number of heads $H = 4$; **WMT14**: $warmup\_step = 8000$, number of heads $H = 8$ similar to Transformer-base model [Vaswani et al., 2017b], $batch\_size = 8192$ tokens.

[7]The model using bold value is achieved a better performance than other values in our experiments.
[8]https://github.com/moses-smt/mosesdecoder

2.4.4 Main Results

We compare the performance of PhraseTransformer with the original Transformer and the previous works' methods on three Semantic Parsing datasets (Table 2.2) and three Machine Translation datasets (Table 2.3).

Table 2.2: Evaluation results using Logic Matching on all datasets. The reported results on Geo are mean and standard deviation values. The values marked (*) mean that the evaluation metric is denotation match that different from others using sentence-level accuracy. This table contains two parts: previous works and our results. . The notations ‡, indicate the corresponding result is statistically significant difference with the baseline (Transformer) in levels $p < 0.01$ [Koehn, 2004].

Model	**Geo**	**Atis**	**MSParS**
Z&C [Zettlemoyer and Collins, 2007]	86.1	84.6	
λ-WASP [Wong and Mooney, 2007]	86.6		
FUBL [Kwiatkowski et al., 2011]	88.6	82.8	
TISP [Zhao and Huang, 2015]	**88.9**	84.2	
Seq2tree [Dong and Lapata, 2016]	87.1	84.6	
Seq2seq+Copy [Jia and Liang, 2016]	89.3*	83.3	
Coarse2Fine [Dong and Lapata, 2018]	88.2	87.7	
DualLearning [Cao et al., 2019]		89.1	
Bert-Sketch [Li et al., 2019]			84.47
Transformer [Ge et al., 2019]			85.68
Transformer (ours)	86.8±0.76	87.7	**85.92**
PhraseTrans.$_{CrossH}$	87.7±0.41‡	89.1	85.43
PhraseTrans.$_{SepH}$	87.9±0.36‡	**90.4**‡	85.72

In the Semantic Parsing task, the experimental results show that all PhraseTransformer models outperform Transformer on two datasets Geo and Atis, especially on Atis with 2.7% Logic Matching in sentence level. On the Geo dataset, we show the learning curve of five runs with different random seeds (Figure 2.6). This result proves that PhraseTransformer clearly outperforms Transformer on the Geo dataset on all checkpoints. On the MSParS dataset, the PhraseTrans.*SepH* model achieves competitive performance with the original Transformer. We observe that this dataset has diverse object names with more than 75% words in vocabulary appearing less than 4 times in training set. One of the challenges of this open-domain dataset is to recognize the object names and their types [Duan, 2019]. For example, the gold logic representation of the input sentence *"what*

Table 2.3: Evaluation results using BLEU score on NMT task on test sets IWSLT14 (de-en) and WMT14 (en-de). The notations ‡, † indicate the corresponding result is statistically significant difference with the baseline (Transformer) in levels $p < 0.01$ and $p < 0.05$, separately [Koehn, 2004].

Model	**IWSLT14**		**IWSLT15**		**WMT14**
	de-en	en-de	en-vi	vi-en	en-de
Transformer [Vaswani et al., 2017b]	34.42	28.35			27.30
MG-SA[Hao et al., 2019]					28.30
BPE-dropout [Provilkov et al., 2020]			**33.27**	**32.99**	28.01
Tree-structured [Nguyen et al., 2020]	35.96	29.47			**28.40**
Transformer (ours)	35.93	29.63	32.20	31.17	27.34
PhraseTrans.$_{CrossH}$	**36.31**‡	**30.06**‡	32.79†	31.84†	27.67†

is birth date for chris pine" is *"(lambda ?x (mso:people.person.date_of_birth chris_pine ?x))"* while the incorrect output of both the Transformer and PhraseTrans.*SepH* is: *"(lambda ?x (mso:biology.organism.date_of_birth chris_ pine ?x))"*. In this case, the semantic parser should have an object information type, *"Chris Pine"* is a person instead of the organism for generating the correct logical form. Therefore, the PhraseTransformer model that supports learning the relation between phrases is hard to show the improvement on this dataset. Compare with the previous works, our model achieves better results on Atis, MSParS and competitive results on the Geo dataset. While our method does not use augmented datasets similarly to Jia and Liang [2016], Ge et al. [2019] or the sketch information [Dong and Lapata, 2018], these results show that our model learns more effectively than the others.

In the Machine Translation task, the PhraseTrans.$_{CrossH}$ improves the performance of the original Transformer on all datasets without external information such as syntactic tree [Nguyen et al., 2020]. Based on the result of the development set of IWSLT14 de-en (detail in the Subsection 2.4.5), because the PhraseTrans.$_{SepH}$ architecture do not show the improvement, we do not evaluate the performance of this architecture on all other MT datasets.

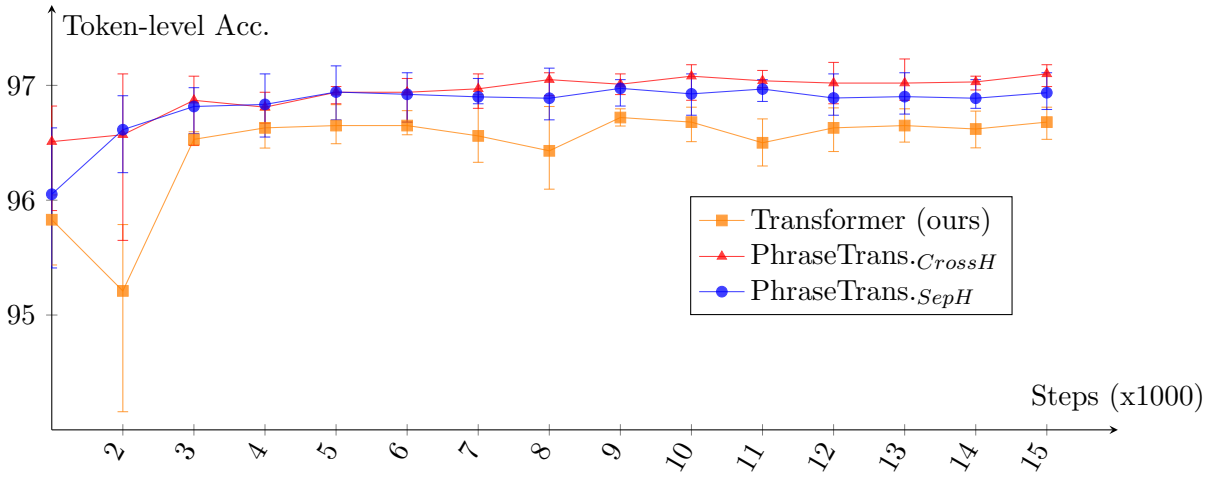


Figure 2.6: Token-level accuracy (min, max and average) of PhraseTransformer and the original Transformer on Geo test set.

2.4.5 Model Variations

We conducted experiments to evaluate the impact of gram sizes on performance on the development set of Atis and MSParS (Table 2.4) and IWSLT14 de-en (Table 2.5) datasets. Besides, we also compare the effectiveness of each PhraseTransformer architecture on both Semantic Parsing and Machine Translation tasks.

Gram Sizes

Based on the result on the Atis dataset, we found that performance increases when applying various gram sizes. The result of both PhraseTrans.$_{SepH}$ and PhraseTrans.$_{CrossH}$ architectures on Atis dataset achieve best performance on setting using three gram sizes 2, 3, 4 (Table 2.4). By using various gram sizes, the PhraseTransformer model can observe different linguistic features in various local context sizes in Multi-head layers. For domain adaptation, the gram sizes can be chosen depending on observing the number of words in meaningful phrases. Using various gram sizes makes PhraseTransformer more generalized. Besides, using LSTM to represent spans on all layers helps PhraseTransformer capture more sequential information than Transformer.

In the Machine Translation task, based on the results on IWSLT14 dev set, we found three settings of gram sizes ({2, 3}, {3} and {4}) achieved competitive performance

Table 2.4: Sentence-level accuracy using logic matching (LM) on two dev sets of Atis and MSParS. The underline values (the best values in each architecture) indicate the setting that will be used in the test set.

Model	**gram sizes**	**#Para.(million)**	**Atis**	**MSParS**
Transformer (origin)		47.1	87.17	86.07
Transformer (larger)		80.0 (+70%)	87.00	
PhraseTrans.$_{SepH}$	$[0;0;0;0;2;2;2;2]$	47.5 (+1%)	88.80	85.99
	$[0;0;0;0;3;3;3;3]$	47.5 (+1%)	88.80	**86.52**
	$[0;0;0;0;2;2;3;3]$	47.5 (+1%)	88.19	85.99
	$[0;0;2;2;3;3;4;4]$	48.3 (+2%)	**89.21**	86.24
PhraseTrans.$_{CrossH}$	$\{2\}$	50.8 (+8%)	87.98	85.50
	$\{3\}$	50.8 (+8%)	87.37	85.39
	$\{4\}$	50.8 (+8%)	87.78	85.48
	$\{2,3\}$	52.4 (+11%)	87.37	85.62
	$\{2,3,4\}$	54.0 (+15%)	88.59	85.72

(Table 2.5). Therefore, we only used these settings to optimize the hyperparameters for other Machine Translation datasets. The solid improvement on all other datasets in five pairs of languages (Section 2.4.4) proved the robustness of our proposed model in domain adaptation.

Table 2.5: BLEU score on IWSLT14 (de-en) dev set.

Model	**gram sizes**	**BLEU**
Transformer (origin)		37.10
PhraseTrans.$_{SepH}$	$[0;0;0;2]$	37.20
	$[0;0;2;3]$	37.15
	$[0;2;3;4]$	37.20
PhraseTrans.$_{CrossH}$	$\{2\}$	37.31
	$\{3\}$	37.47
	$\{4\}$	**37.58**
	$\{2,3\}$	37.56
	$\{2,3,4\}$	37.44

Model Size

We compare the number of parameters in PhraseTrans.$_{SepH}$ and PhraseTrans.$_{CrossH}$ with the original Transformer of experiments on MSParS (Table 2.4). The number of parameters of the PhraseTrans.$_{SepH}$ is slightly increased (less than 2%) when compared with

the original Transformer while the PhraseTrans.$_{CrossH}$ increase arround 15% with three different gram sizes. The reason is that in the Phrase Trans.$_{CrossH}$, the n-gram LSTM is applied with a hidden size larger two times than PhraseTrans.$_{SepH}$ by zip method on the query and key vectors. Besides, we also increase the hidden side of the original Transformer to a large setting, which increases nearly two times (+70%) of the base model size, however, the result does not improve. This result proves our model architecture is efficient.

Computation time

We compare the training speed between the Transformer and PhraseTransformer on the Atis dataset (Table 2.6). This experiment is conducted on an NVIDIA® Tesla® P100 16G with batch size of 4096 tokens. Compared with the original Transformer, the training speed of PhraseTrans.$_{SepH}$ model is decreased 15% and PhraseTrans.$_{CrossH}$ model is decreased 35%. The computation cost of PhraseTrans.$_{CrossH}$ is larger than PhraseTrans.$_{SepH}$ because with each gram size in *CrossH* architecture, phrase modeling mechanism is applied on all heads, while in *SepH* architecture, each gram size is applied with corresponding heads following hyper-parameter setting instead of all heads.

In fact, although we used LSTM on heads of Multi-head layers, the computation time is not dependent on the length of sentence because we can forward and backward all n-grams of all sentences in a minibatch at the same time. Therefore, the computation time is more dependent on the gram size (in this case, the maximum gram size is 4).

Table 2.6: Computation time comparison between our PhraseTransformer and the original Transformer on Atis dataset (K indicate thousands words processed per second).

Model	**Training Speed**	**Inferring Speed**
Transformer (ours)	8.0 K	15.7 K
PhraseTrans.$_{SepH}$	6.8 K (-15 %)	11.8 K (-25%)
PhraseTrans.$_{CrossH}$	5.2 K (-35 %)	10.6 K (-32%)

Phrase Intergration

Finally, we observe the effectiveness of two PhraseTransformer architectures on two tasks Semantic Parsing and Machine Translation. In the Semantic Parsing task (Table 2.4), the results showed that the PhraseTrans.$_{SepH}$ are more appropriate than PhraseTrans.$_{CrossH}$. On both MSParS and Atis datasets, the architecture PhraseTrans.$_{SepH}$ achieves performance better than PhraseTrans.$_{CrossH}$ in almost settings of gram sizes. In the Machine Translation task (Table 2.5), the architecture PhraseTrans.$_{CrossH}$ achieved performance better than PhraseTrans.$_{SepH}$ architecture in all settings of gram sizes. We argue that the reason comes from the characteristics of data in two tasks. In the Semantic Parsing task, the vocabulary of the dataset is small for a special domain (Geo, Atis) or in a set of limited question types (MSParS), and the content of these datasets is the combination of sub-conditions together or paraphrasing them. Therefore, the phrase representation and relation between phrases play an important role to generate the correct logical form. In the Machine Translation task, the content of the dataset is more general with a vocabulary larger than the Semantic Parsing dataset. By replacing the word representations with phrase representations, some important information of single words is missed. In PhraseTrans.$_{CrossH}$ architecture, we keep all original representations of single words and append phrase representations as a context that improves the meaning representation of each word.

In addition, we conducted experiments to study the affect of different methods incorporating local context information into word representation. For mathematical, Equation 2.11 is replaced by following formulas:

$$\boldsymbol{q}'_i, \boldsymbol{k}'_i, \boldsymbol{v}'_i = \mathcal{F}(\boldsymbol{q}_i, \text{Phrase}(\boldsymbol{q}_i, n_i)), \mathcal{F}(\boldsymbol{k}_i, \text{Phrase}(\boldsymbol{k}_i, n_i)), \mathcal{F}(\boldsymbol{v}_i, \text{Phrase}(\boldsymbol{v}_i, n_i)) \quad (2.20)$$

where $\mathcal{F}$ is the incorporating function chosen from {Min, Max, Average}. After that, new query, key, value vectors is forwarded to the next layer. We showed the result of this ablation study on Atis dev set in Table 2.7. The result showed that with all different incorporating methods, PhraseTransfomrer is better when compared with the original

Transformer (87.17% sentence accuracy). Besides, the results of different incorporating methods are quite competitive together.

Table 2.7: Variants model using the different methods incorporating local context into word representation on Atis dev set.

Model	Sent. Accuracy
PhraseTrans.$_{SepH}$	**89.21**
PhraseTrans.$_{SepH}$ ($\mathcal{F}$ = Average)	88.59
PhraseTrans.$_{SepH}$ ($\mathcal{F}$ = Max)	88.39
PhraseTrans.$_{SepH}$ ($\mathcal{F}$ = Min)	88.59

2.5 Result Analysis

2.5.1 Examples of Improvement

We analyze examples that our PhraseTransformer improved over the original Transformer (Table 2.8). The improvement can be grouped into three types of errors: (1) 46.2% the errors are caused by Transformer confusing the role of entities name such as *"ci2"* and *"ci0"* (row 1 on Table 2.8); (2) 27.3% missing semantic components such as *"(round_trip $0)"* (row 2); (3) 27.3% wrong in predicate name of logic component (row 3). In addition, we also show an incorrect prediction of both models in row 4. In this sample, the Phrase-Transformer improved *from*, *to*, and *round_trip* sub-expressions, which are confused by Transformer. Although there is still a different sub-expression (*fare*) with the gold logic, the meaning of both is the same. These results proved that the superior of the Phrase-Transformer over the Transformer is due to the improvement of the capacity of capturing local context information as well as the meaning relation between phrases.

2.5.2 Length Analysis

To analyze the impact of input sequence length on the performance of Transformer and PhraseTransformer, following the previous works [Bahdanau et al., 2015, Xu et al., 2020b] we conducted the experiment by splitting and evaluating the test sets of the IWSLT14

Table 2.8: Examples that frequent incorrect predictions of Transformer, are improved in PhraseTransformer on the Atis test set. The red tokens indicate the wrong token predictions of the semantic parser. The notations (✓, ✗) indicate the correct and incorrect prediction at the sentence level, respectively.

Sentence		*what are the flight **from ci1 to ci2** that **stop in ci0***
Gold LF		(lambda $0 e (and (flight $0) (from $0 ci1) (to $0 ci2) (stop $0 ci0)))
Transformer	✗	(lambda $0 e (and (flight $0) (from $0 ci1) (to $0 ci0) (stop $0 ci2)))
PhraseTransformer	✓	(lambda $0 e (and (flight $0) (from $0 ci1) (to $0 **ci2**) (stop $0 **ci0**)))
Sentence		*give me the cheapest **round trip** flight from ci0 to ci1 around mn0 dn0*
Gold LF		(argmin $0 (and (flight $0) ... (month $0 mn0) (round_trip $0)) (fare $0))
Transformer	✗	(argmin $0 (and (flight $0) ... (month $0 mn0)) (fare $0))
PhraseTransformer	✓	(argmin $0 (and (flight $0) ... (month $0 mn0) **(round_trip $0)**) (fare $0))
Sentence		*show me the airport **servic by al0***
Gold LF		(lambda $0 e (and (airport $0) (services al0 $0)))
Transformer	✗	(lambda $0 e (and (airport $0) (airline $0 al0)))
PhraseTransformer	✓	(lambda $0 e (and (airport $0) (**services al0 $0**)))
Sentence		*give me the flight and **fare** for a **trip to ci0 from ci1** on da0*
Gold LF		(lambda $0 e (exists $1 (and (flight $0) (from $0 ci1) (to $0 ci0) (day $0 da0) (fare $0 $1))))
Transformer	✗	(lambda $0 e (exists $1 (and (flight $0) (round_trip $0) (from $0 ci0) (to $0 ci1) (day $0 da0) (= (fare $0) $1))))
PhraseTransformer	✗	(lambda $0 e (exists $1 (and (flight $0) **(from $0 ci1) (to $0 ci0)** (day $0 da0) (= (fare $0) $1))))

de-en and Atis datasets into groups of different lengths (Figures 2.7, 2.8). These results show that our proposed model beat the vanilla Transformer in all groups of sentence length, which show our model generalization abilities. In addition, the improvement of the proposed model is more clearly in the long sentences on both Semantic Parsing and Machine Translation tasks. The PhraseTransformer architecture work effectively in all groups of sentence length because the meaning representations of single words are improved by the phrase representations as local context information.

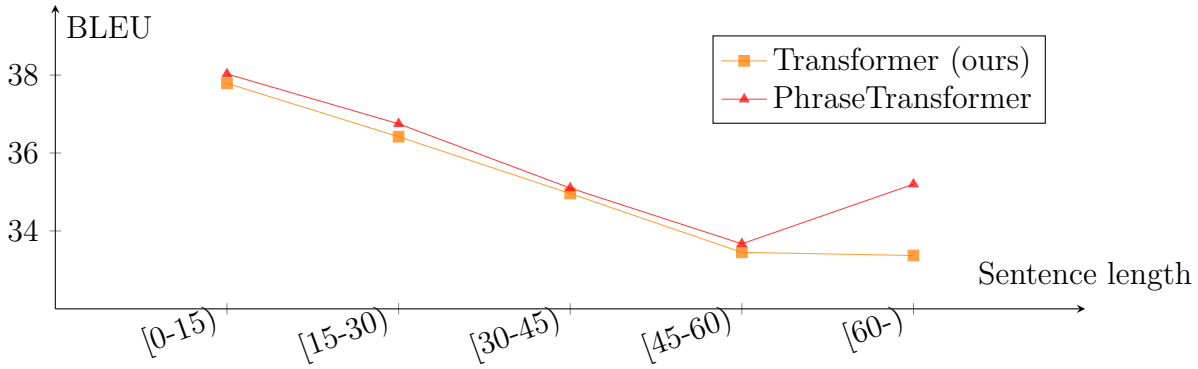


Figure 2.7: BLEU scores of PhraseTransformer (best model on dev set) and the Transformer on IWSLT14 de-en test set with respect to the source sentence length. The number of samples in each sub-set is 3176, 2499, 760, 228 and 87, respectively.

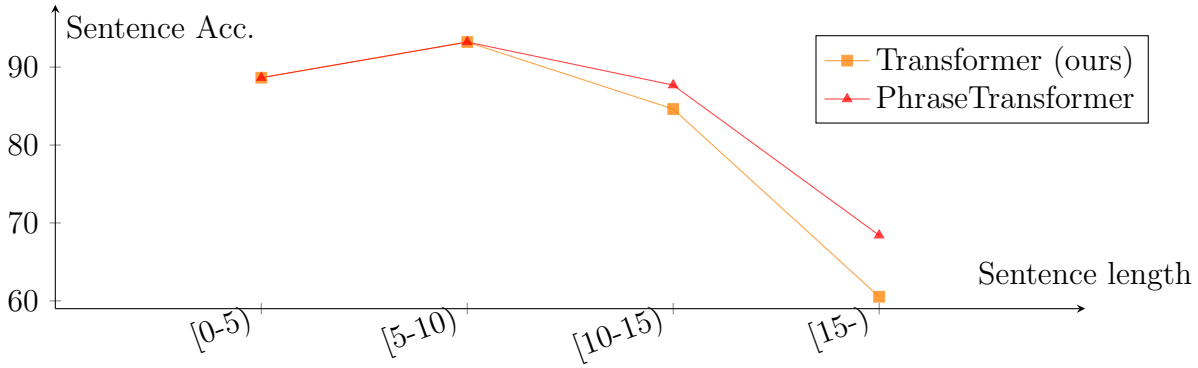


Figure 2.8: Performance comparison of PhraseTransformer and the Transformer on Atis test set with respect to the source sentence length. The number of samples in each sub-set is 44, 236, 130, and 38, respectively.

2.5.3 Self-Awareness

Attention Alignment We inspect the information learned in PhraseTransformer in Attention layers (Figure 2.9). We observe that PhraseTransformer could represent at-

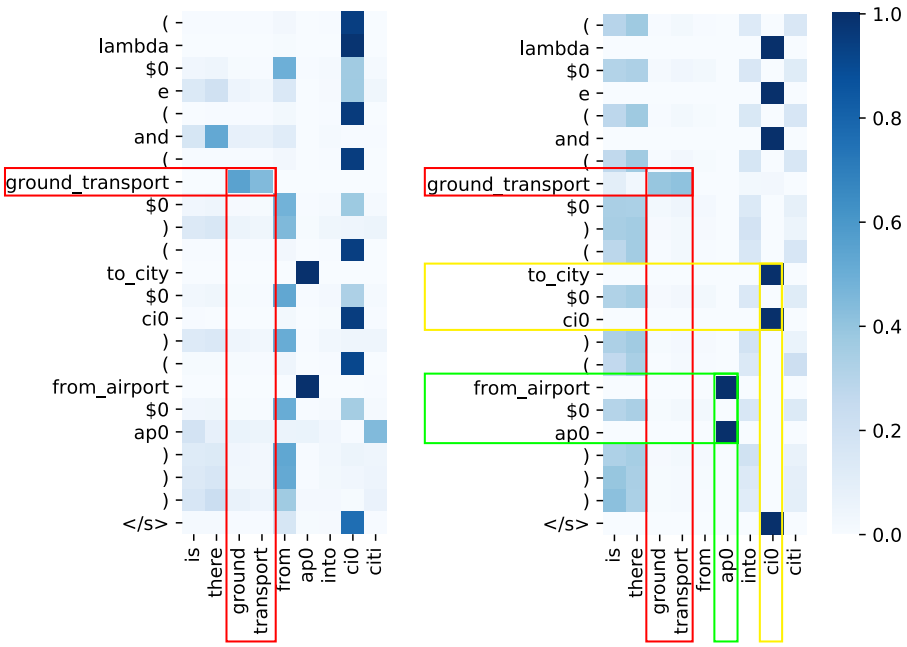
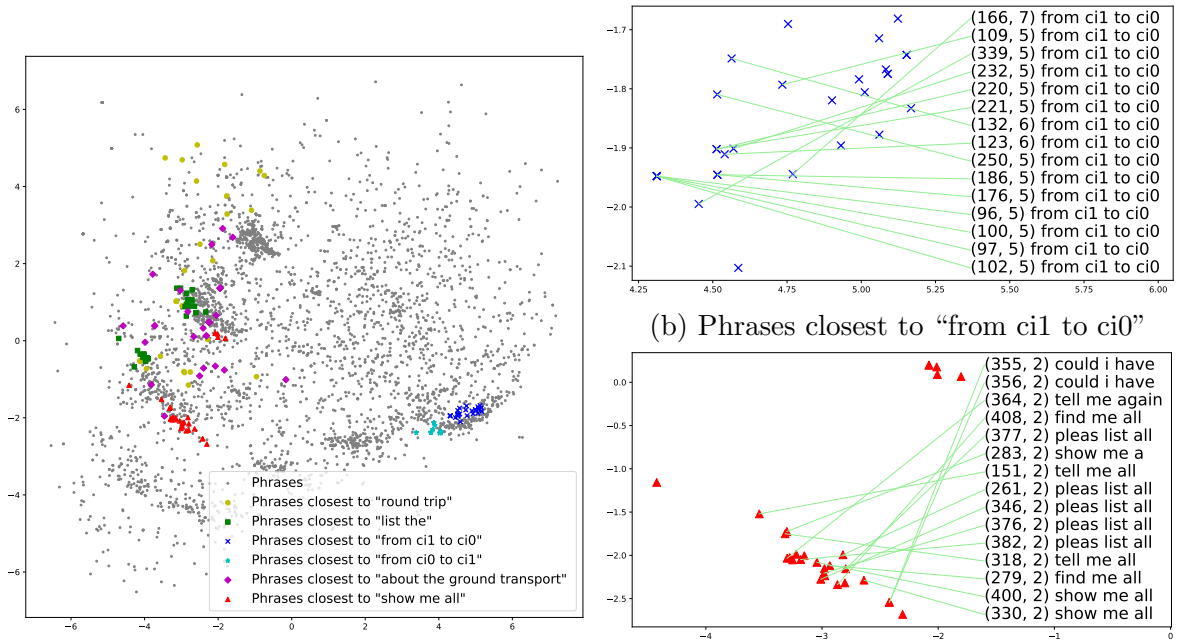


Figure 2.9: Heatmap visualization of Encoder-Decoder Attention of the original Transformer (left) and PhraseTransformer (right). Considering one row, the value in each column is corresponding to the rate of the attention of token in LF to the word in the sentence.

tention information more clearly than Transformer. In both two models, the token *ground_transport* in LF is aligned correctly to phrase *"ground transport"* in the sentence (red alignments). In PhraseTransformer, tokens *to_city*, *from_airport* are also correctly aligned to the corresponding words *"ap0"*, *"ci0"* in the sentence (green and yellow alignments) because these word vectors probable to capture local context better than Transformer. Besides, all tokens decoded by PhraseTransformer paid the same attention to other words that is not key information, such as *"is there"*, *"into"*, *"citi"*. These evidences is positive signals showing that the self-awareness of PhraseTransformer is better than Transformer.

Meaning Phrase In this experiment, we explore the natural language understanding capacity of our PhraseTransformer. We use the Principal Component Analysis (PCA) method to visualize the similarity of phrases in PhraseTransformer best setting on Atis dataset (Figure 2.10) by using hidden state of heads 7, and 8 (the vector $[\boldsymbol{q}'_7; \boldsymbol{q}'_8]$ where $\boldsymbol{q}'_i$ from Equation 2.11). We also highlight 30 closest points (the distance using Cosine distance) to the particular phrase carrying key information such as *"round trip"*, *"from ci1 to ci0"*. Besides, we also visualize the vector of words ($[\boldsymbol{q}_7; \boldsymbol{q}_8]$ where $\boldsymbol{q}_i$ from Equation 2.1) to show the lacked local context information of word vectors in the original Transformer in Figure 2.11b.



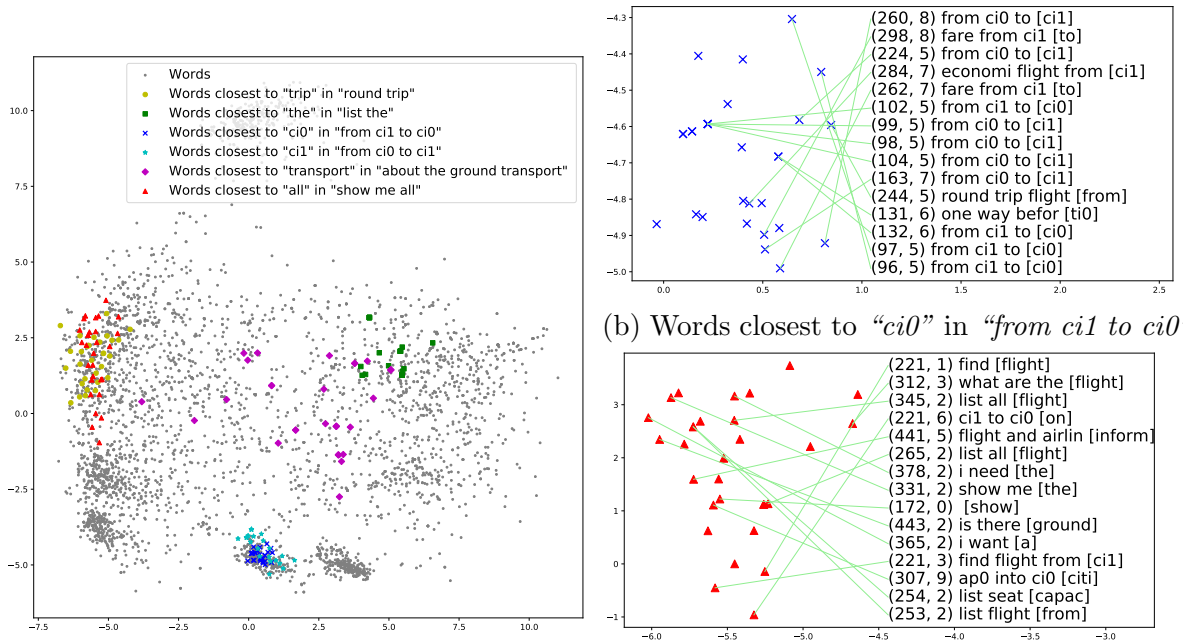
(a) All phrases

(b) Phrases closest to "from ci1 to ci0"

(c) Phrases closest to "show me all"

Figure 2.10: Figure a draws the representing vector of phrases in Selft-Attention layer of PhraseTransformer using PCA on Atis test set. Figures b, c are zoomed-in view of the blue and red clusters. The labels are annotated for each point show the information of the phrase corresponding to point following the template *(sentence_id, pharse_position) phrase_content*.

Considering two frequent phrases *"from ci1 to ci0"* and *"from ci0 to ci1"* of Atis dataset on PhraseTransformer (Figure 2.10a), the phrases closest to two phrases concentrate on blue and cyan clusters. These two clusters are closest to each other but separate without overlapping. Compare with the original Transformer (Figure 2.11a), these clusters are



(a) All words

(b) Words closest to *"ci0"* in *"from ci1 to ci0"*

(c) Words closest to *"all"* in *"show me all"*

Figure 2.11: Figure a draws the representing vector of words in Selft-Attention layer of the original Transformer using PCA on Atis test set. Figures b, c are zoomed-in view of the blue and cyan clusters. The labels are annotated for each point in two figures show the information of the word corresponding to point following the template *(sentence_id, word_position) phrase_context [considering_word]*.

overlapped together. This feature helps the decoder decode different semantic components such as *(from $0 ci0) (to $ 0 ci1)* and *(from $0 ci1) (to $0 ci0)*. We argue that, this is the useful sequence characteristic that LSTM architecture contribute to the original Transformer architecture. In other aspect, Figure 2.10b shows that the phrase *"from ci1 to ci0"* is represented by the similar vectors in various contexts as well as positions. For example, this phrase in Atis data sentence 175 *"show me nonstop flight from ci1 to ci0"* has the same meaning in sentence 339 *"a flight from ci1 to ci0 arriv between ti0 and ti1"*.

In Figure 2.10c, there are many different phrases having the same meaning that the model finds out, such as *"could i have"*, *"tell me again"*, *"find me all"* or the phrases closest to *"list the"* and *"show me all"* in Figure 2.10a. These phrases do not contain query information, which is the robust feature of human natural language, this is an evidence that the model can learn complicated characteristics of natural language.

2.5.4 Encoder Self Attention

Figure 2.12 shows the difference between heads in Self-Attention Encoder of PhraseTransformer using PhraseTransformer architecture. The self-attention in heads that do not use n_gramLSTM is more incoherent than other heads. For example, in head 1, almost words

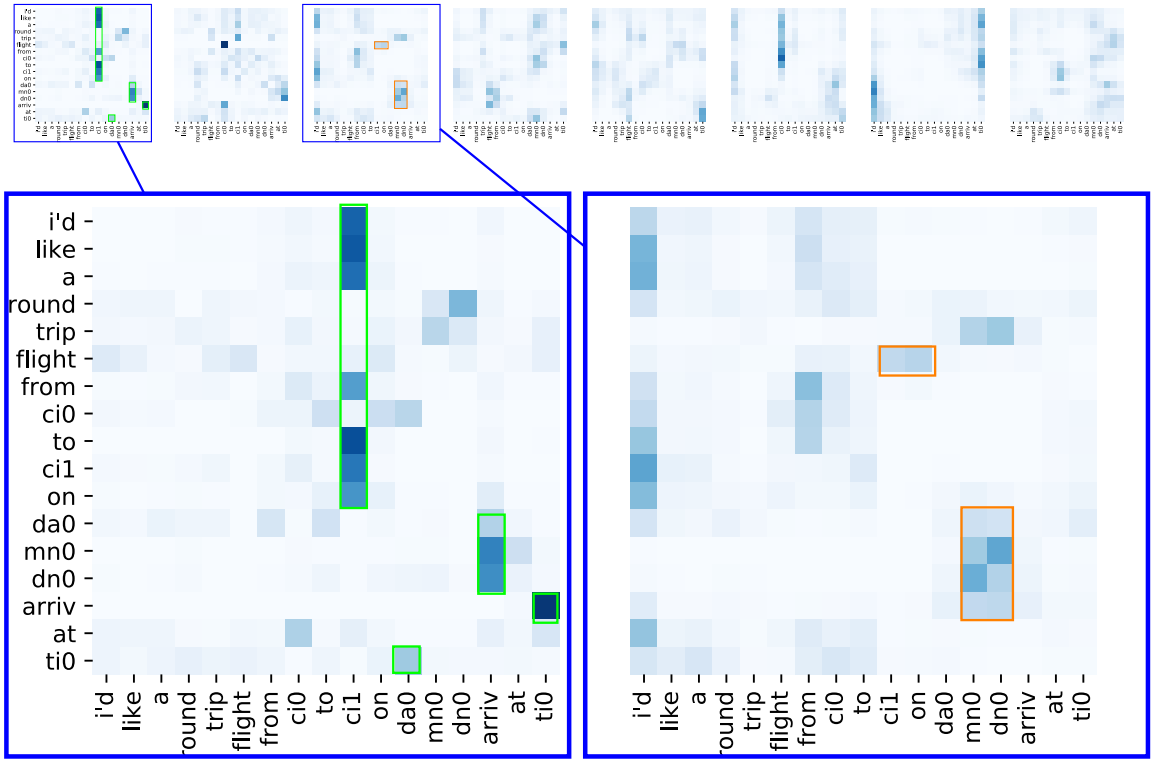


Figure 2.12: Heatmap visualization of Attention. This figure shows Self-Attention in 8 heads of the last PhraseTransformer Encoder layer. Two blue rectangles are zoomed-in separately of head 1 (not use n_gramLSTM), head 3 (use 2_gramLSTM).

in query focus on *"ci1"* and the other words are paid attention is key information words such as *"da0"*, *"arriv"*, *"ti0"* (the green rectangles). From head 3 to 8, the attention focuses on the separated clusters, which shows that model learned the dependencies of the phrases instead of the single words. On these heads, the attentions are usually between groups important words such as *"flight"* with *"ci1 on"*, *"da0 nm0 dn0 arriv"* with *"nm0 dn0"* (the orange rectangles).

2.6 Conclusion

In this work, we proposed a novel model named PhraseTransformer that can improve the performance of the Transformer in semantic parsing task and NMT task. We enhance Transformer Encoder to improve the representing ability of the detailed meaning of a sentence based on learning the phrase dependencies. In the methods using Neural Network, this model obtains SOTA results on the Atis and MSParS datasets and achieves a competitive result with the SOTA in other datasets. We also conducted experiments to compare with Transformer and show the improvement of self-attention in PhraseTransformer architecture. In future work, we would like to extract more information about the relationship between words or phrases leveraging this architecture and investigate on how to inject prior knowledge to improve it. We believe that this architecture can be widely applied in many problems using sequence to sequence models such as neural machine translation and abstract text summarization.

Chapter 3

Class Imbalance in Spoken Language Understanding

In the semantic parsing field, Spoken Language Understanding (SLU) is one of the tasks widely applied in realistic applications in recent years. In the success of the pre-trained BERT model, NLU is addressed by Intent Classification and Slot Filling task with significant improvement performance. However, classed imbalance problem in NLU has not been carefully investigated, while this problem in Semantic Parsing datasets is frequent. Therefore, this work focuses on diminishing this problem. We proposed a BERT-based architecture named JointBERT Classify Anonymous Entity (JointBERT-CAE) that improves the performance of the system on three Semantic Parsing datasets ATIS, Snips, ATIS Vietnamese, and a well-known Named Entity Recognize (NER) dataset CoNLL2003. In JointBERT-CAE architecture, we use multitask joint-learning to split conventional Slot Filling task into two sub-task, detect *Anonymous Entity* by Sequence tagging and *Classify* recognized anonymous entities tasks. The experimental results show the solid improvement of JointBERT-CAE when compared with BERT on all datasets, as well as the wide applicable capacity to other NLP tasks using the Sequence Tagging technique.

3.1 Introduction

Nowadays, with the rapid development of virtual assistants and dialog systems such as Google Home, Amazon Alexa, numerous researches investigate the SLU task which is the core component of smart speakers. The SLU task is typically addressed by two essential sub-tasks that include Intent Prediction (ID) and Slot Filling (SF) tasks [Tur and De Mori, 2011]. Recently, the impressive improvements [Chen et al., 2019, Castellucci et al., 2019b] are largely based on the success of pre-trained language models with little fine-tuning (e.g. BERT [Devlin et al., 2019a]).

However, most previous works have not considered the class imbalance problem in the Slot Filling task. Based on our primary analysis about the distribution of entity types in two well-known SLU datasets: Snips [Coucke et al., 2018] and ATIS [Hemphill et al., 1990] (Figure 3.1), we found that the class imbalance problem in these datasets is highly critical, especial on ATIS. Indeed, the target of the SF task is to extract slot

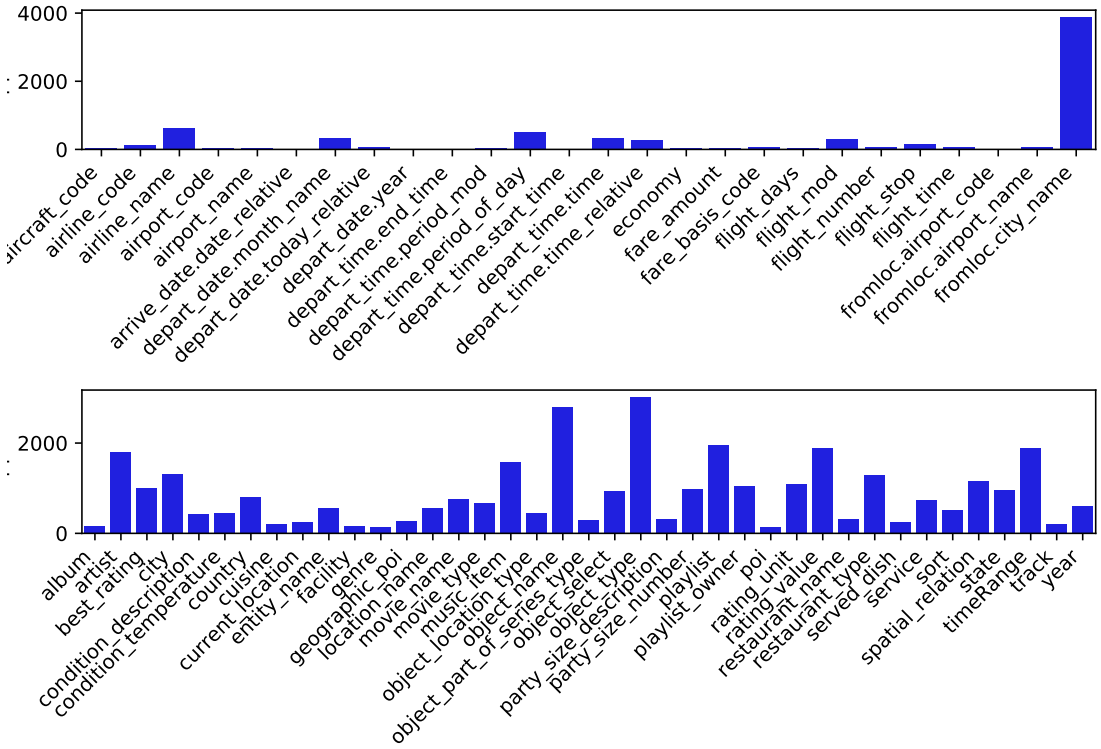


Figure 3.1: Distribution of Slot classes in ATIS (top) and Snips (bottom) datasets. For the space limitation, some Slot classes in ATIS are ignored. In the graph, x, y denotes the Slot class name and the number of instances.

information that usually is the text span in the input natural sentence. Similar to the Named Entity Recognition (NER) task in Natural Language Processing (NLP), this task is solved by sequence labeling technique with BIO schema [Wang et al., 2018, Chen et al., 2019, Castellucci et al., 2019b, Qin et al., 2021b]. In the NER area, the class imbalance problem is also investigated by previous works [Wang et al., 2021, Grancharova et al., 2020]. In the SLU task, given a sentence, the required output is the intent and slots information of the sentence. With each kind of intent, the type of slot information may differ, therefore, the number of slot information types is ordinarily high. For example, there are 79 different Slot types in ATIS. Besides, not only the imbalance among classes of entities but also the imbalance between positive and negative words is also important. In detail, there is a small number is positive words that are inside entity names, while there is a large number of negative words which are outside entity span. Both kinds of imbalance affect the performance of the SF task as well as the overall system [Grancharova et al., 2020].

The previous works [Grancharova et al., 2020, Wang et al., 2021, Li et al., 2020a] related to the class imbalance almost show the solutions on the NER task. [Grancharova et al., 2020] proposed a re-sampling data method to diminish this problem by duplicating the samples of less occurrence class. This method is proven to work well on Stockholm EPR PHI Corpus [Grancharova et al., 2020] which has the most common class larger 24 times than the smallest class. However, the authors choose the threshold of oversampling size manually without explanation. Besides, in the Semantic Parsing task, especially on Atis, the most common class is larger 400 times than the smallest class, which leads to the number of oversampling that might be larger than the size of the original dataset. Recently, [Wang et al., 2021, Li et al., 2020a] proposed the approaches based on Machine Reading Comprehension-based (MRC) to solve the NER task. These works replace each class label with its natural description and pair it with the original sentence to make the input of the MRC model for entity position detection. However, this approach increases the data training size by the number of entity classes ($|C|$) times [Wang et al., 2021]. While the number of entity classes in Semantic Parsing datasets is a large number (e.g.

the number of entity classes of ATIS is 79). This characteristic is the main factor that makes difference between the NER and the Slot Filling tasks in SLU.

In this work, we introduce a mechanism named Classify Anonymous Entity (CAE) inspired from the previous works using MRC-based architecture [Wang et al., 2021, Li et al., 2020a] to deal with Slot Filling in SLU task. In our proposed mechanism, we also split the original entity recognition (or slot detection) process into two sub-tasks (Figure 3.2): determine the span of the entity or slot information as an anonymous entity, and classify the recognized anonymous entities into the related class. However, the main difference

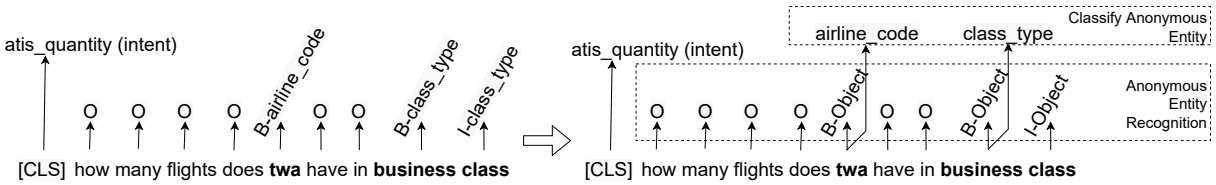


Figure 3.2: Comparison of SLU task using Joint ID and SF task between original approach (left) and our proposed approach using CAE mechanism (right).

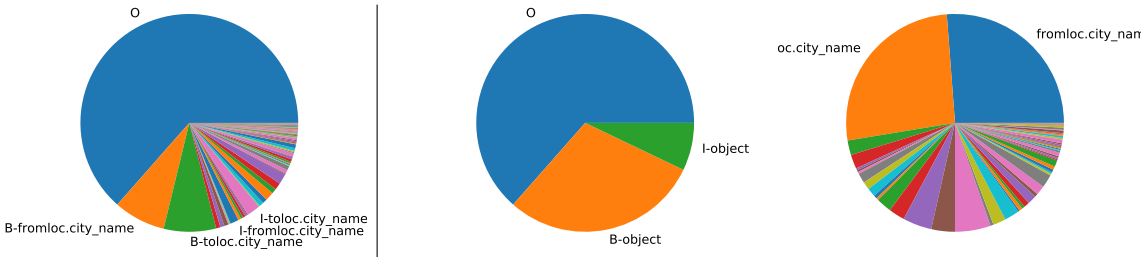
between our mechanism and previous works using MRC-based architecture is that the span entity detection and classify entity are done by sequence tagging architecture. In our method, it is unnecessary to expand artificial samples in the training process. Besides, the proposed mechanism diminishes the class imbalance between entity classes with the outside entity (`O`) class. More detail, we design a new entity class called `Object` for span entity recognition and replace all original entity classes labels by new entity `Object` label. For example, `B-PER`, `B-ORG` labels are replaced by `B-Object` label, and apply by similar way to the `I-*` labels. After that, each anonymous entity is classified into the related entity class in the second step.

Class-imbalance Measuring. Follow the information theory [Shannon, 2001], Entropy (H) is the measure of uncertainty of a random variable or the amount of information required to describe a variable. Therefore, we use the Entropy measurement to compute the balance degree of the slots classes that classifiers need to process in the original

approach and our approach using CAE mechanism.

$$\text{balance degree} = -\frac{1}{\log(k)} * \sum_{1}^{k} \frac{c_i}{n} \log(\frac{c_i}{n}) \tag{3.1}$$

where k is the slots class number, n is the number of example given in whole dataset, c_i is the number examples of slot class ith in the dataset. For example, in the best balancing case, the slot classes have the same number of example ($c_i = n/k$ examples), *balance degree* is 1; in the worst imbalance case, the number of classes is 1 ($k = 1$), *balance degree* is 0. More specifically, we show the slot classes distribution and *balance degree* in Figure 3.3 in the original approach and the approach using the CAE mechanism. In our mechanism, the first classifier only deals with three classes to detect the anonymous entity span, so the balance degree is high with 0.76 scores; the second classifier only considers to begin of slot classes `B-*` (skip outside entity class - `O` and inside entity class `I-*`) so the balance degree is also improved with 0.62 scores. Therefore, our mechanism can diminish the slots class imbalance in the Slot Filling sub-task of the SLU system.



(a) Original slot distribution, balance degree is 0.38

(b) Slot distribution of two classifier in CAE mechanism with balance degrees are 0.76 and 0.62, respectively.

Figure 3.3: Comparison of slot class distribution between original approach (Figure 3.3a) and our approach using CAE mechanism in Atis dataset (Figure 3.3b).

Briefly, our contributions are summarized as follows.

- We propose a simple yet effective mechanism, CAE, to handle the class imbalance problem in the Slot Filling task of SLU as well as the NER task.
- Our experimental results show that our proposed model improves the performance of the Slot Filling task using the F1 score leads to improve performance of the

overall system on two Semantic Parsing datasets Snips, ATIS, and well-known NER dataset, CoNLL 2003, compared with the original approach.

- The proposed model achieves new SOTA performance on ATIS Vietnamese dataset [Dao et al., 2021] with a 1.4 F1 score improvement.

3.2 Related Work

3.2.1 SLU task

The SLU task using the Deep learning model has been attracted by numerous works for a long time [Xu and Sarikaya, 2013, Wang et al., 2018, Chen et al., 2019, Castellucci et al., 2019b, Qin et al., 2021b, 2020, He et al., 2021, Qin et al., 2021a]. In the first development period of this task, two sub-tasks ID and SF are typically addressed by Convolutions Neural Network (CNN) [Xu and Sarikaya, 2013] and Long Short-Term Memory (LSTM) Network [Ravuri and Stolcke, 2015, Wang et al., 2018]. In the success of pre-trained language using the Self-Attentive model [Vaswani et al., 2017a, Devlin et al., 2019a] and two-stage training fashion in the last few years, the performance of the SLU system is substantially improved in many recent works [Chen et al., 2019, Castellucci et al., 2019b]. Together with the strength of pre-trained language models, many approaches put much more effort into improving the overall systems. In particular, [He et al., 2021] focuses on incorporating external knowledge (e.g. WordNet) to enhance the performance of SF sub-task, especially boosting out-of-vocab words recognition. [Goo et al., 2018, Li et al., 2018, Qin et al., 2019, 2020] introduces architectures targeted to intent-slot interaction, and [Qin et al., 2020] shows the advancement on the SLU multi-intent task. Besides, [Qin et al., 2021a] also focuses on co-interaction between intent and slot information. This work is inspired by vanilla Transformer with intent while slot information is considered as a query and key component in the self-attention mechanism. Difference from the previous works, our work focus on the class imbalance of the Slot Filling task. To our best knowledge, we are the first to investigate this problem in the SLU task.

3.2.2 Class Imbalance in Sequence Labeling

Based on the primary analysis shown in the introduction section, we found that the imbalance class in the Slot Filling of SLU is quite critical. To deal with this problem, there are lots of approaches in previous works, especially in the NER task [Li et al., 2020b,a, Grancharova et al., 2020]. [Li et al., 2020b] introduces a dynamic adjusted-weight loss function that reflects the importance of easy-negative examples in training data. [Grancharova et al., 2020] proposes methods for re-sampling training data based on the distribution of entity classes, especially oversampling approach. The works [Wang et al., 2021, Li et al., 2020a] focus on adapting MRC architecture to deal with the NER task. By this approach, the model can ignore the imbalance between positive (inside entity) and negative (outside entity) words, however, it is a bias into imbalance among entity classes and increases the number of training examples. In our proposed model using the CAE mechanism, the imbalance among positive and negative words is diminished by combining all original entity classes into a special object entity class (`Object`). Besides, this mechanism in technical also abates the imbalance among entity classes by removing meta labels (`B-*`, `I-*`) in each entity class. Therefore, the number of training samples has remained and the imbalance problem is partly addressed by our delicate consideration.

3.3 Methodology

In this section, we describe the detail of the competitive baselines, the oversampling mechanism [Grancharova et al., 2020] and our proposed model using CAE mechanism incorporating with Conditional Random Field (CRF) layer.

3.3.1 Baseline Model

BERT model. The architecture of this model is the combination of multiple Transformer Encoders [Vaswani et al., 2017a] layers. In each Encoder layer, the major component to extract and digest linguistic features is the Self-Attention layer that learns the

long-range dependencies between the pairs of words in a sentence. Given the input is the natural sentence ($\boldsymbol{s} = \{w_k\}_1^{|s|}$ where $|s|$ is the number of words[1]), by using BERT model, we get the hidden vector representation of each word ($\boldsymbol{h}_k$). For the classification task, the authors Devlin et al. [2019a] introduce a simple method adding a special token (*[CLS]*) into the input sentence and using the hidden vector of this token for sentence representation.

$$\boldsymbol{h}^{[CLS]}, \boldsymbol{h}_k^{word} = \text{BERT}(\boldsymbol{s}) \tag{3.2}$$

where k is the word index in the sentence.

JointBERT model. We follow the previous work [Chen et al., 2019] to handle sub-tasks Intent Detection (ID) and Slot Filling (SF) by joint learning all sub-tasks together. For ID task, hidden vector of [CLS] token ($\boldsymbol{h}^{[CLS]}$) is forwarded to Dense layer to reduce dimension and processed by a *softmax* function to get intent probabilities.

$$\boldsymbol{y}^{ID} = \text{softmax}(\boldsymbol{W}^{ID}\boldsymbol{h}^{[CLS]} + \boldsymbol{b}^{ID}) \tag{3.3}$$

where $\boldsymbol{W}^{ID}, \boldsymbol{b}^{ID}$ is learnable parameters. For SF task, after we get the hidden vector of words ($\boldsymbol{h}_i^{word}$) encoded by BERT model, these vectors are also forwarded to Dense and *softmax* layers.

$$\boldsymbol{y}_k^{SF} = \text{softmax}(\boldsymbol{W}^{SF}\boldsymbol{h}_k^{word} + \boldsymbol{b}^{SF}) \tag{3.4}$$

where $\boldsymbol{W}^{SF}, \boldsymbol{b}^{SF}$ is learnable parameters. Besides, if a word in the sentence is split into sub-words (by the BERT Tokenizer module), only the first sub-word will be used for the whole original word representation for Slot label prediction. Finally, for the joint training process, the objective loss function is computed by the weighted sum of the Cross-Entropy[2]

[1] We use "word" to simplify, however, in the practice, it should be sub-words split by a BERT Tokenizer module (e.g. WordPiece).

[2] Use mean reduction in implementation.

losses of SF and ID sub-tasks.

$$\mathcal{L} = \text{CrossEntropy}(\boldsymbol{y}^{ID}, \mathbf{y}^{gID}) + \lambda \times \sum_{k=1}^{|s|} \text{CrossEntropy}(\boldsymbol{y}_k^{SF}, \mathbf{y}_k^{gSF}) \tag{3.5}$$

where λ is the hyper-parameters to adjust the strength of SF loss; $\mathbf{y}^{g*}$ is the gold labels from SLU datasets.

3.3.2 Oversampled data

The result from the previous work [Grancharova et al., 2020] shows that oversampling technique on NER tasks can improve the performance of the overall system for imbalanced datasets. The target of this mechanism is to duplicate the samples of minority classes and endeavor the balance among entity classes in training data. To this end, we construct a threshold r is the ratio of samples in minority entity classes ($0 < r < 1$) that need to be reached when comparing with the largest entity class. After that, the sentences that contained labels of minority classes are randomly selected until the ratio of all these entity classes reaches the threshold r.

3.3.3 Proposed model

JointBERT-CAE model.

In this architecture, we use two different classifiers for the SF task (Figure 3.4). The first one is used for anonymous entity span recognition while the second is for related entity (slot) classification. Especially, the second classifier only considers positive words which are in the recognized anonymous entity span. To this end, the second classifier does not face the imbalance problem between positive (inside entity) and negative (outside entity) words. For the ID task, we follow the baseline model architecture. Mathematically,

Equation 3.4 is replaced by the following formulas:

$$\mathbf{y}_k^{SF1} = \text{softmax}(\mathbf{W}^{SF1}\mathbf{h}_k^{word} + \mathbf{b}^{SF1}) \tag{3.6}$$

$$\mathbf{y}_k^{SF2} = \text{softmax}(\mathbf{W}^{SF2}\mathbf{h}_k^{wordEntity} + \mathbf{b}^{SF2}) \tag{3.7}$$

where $\mathbf{W}^*, \mathbf{b}^*$ is learnable parameters; $\mathbf{h}^{wordEntity}$ is the hidden states of positive words (the bold words in the example of Figure 3.4). Finally, the objective loss function is also computed by combination of the weighted Cross-Entropy losses of SF and ID sub-tasks for the joint training process.

$$\begin{aligned}\mathcal{L} = \text{CrossEntropy}(\boldsymbol{y}^{ID}, \mathbf{y}^{gID}) + \lambda \times \sum_{k=1}^{|s|} \Big(&\text{CrossEntropy}(\boldsymbol{y}_k^{SF1}, \mathbf{y}_k^{gSF1}) \\ &+ f_k \times \text{CrossEntropy}(\boldsymbol{y}_k^{SF2}, \mathbf{y}_k^{gSF2})\Big)\end{aligned} \tag{3.8}$$

where $\mathbf{y}^{g*}$ is the gold labels from SLU datasets; f_k is the flag storing positive ($f_k = 1$) or negative ($f_k = 0$) word information.

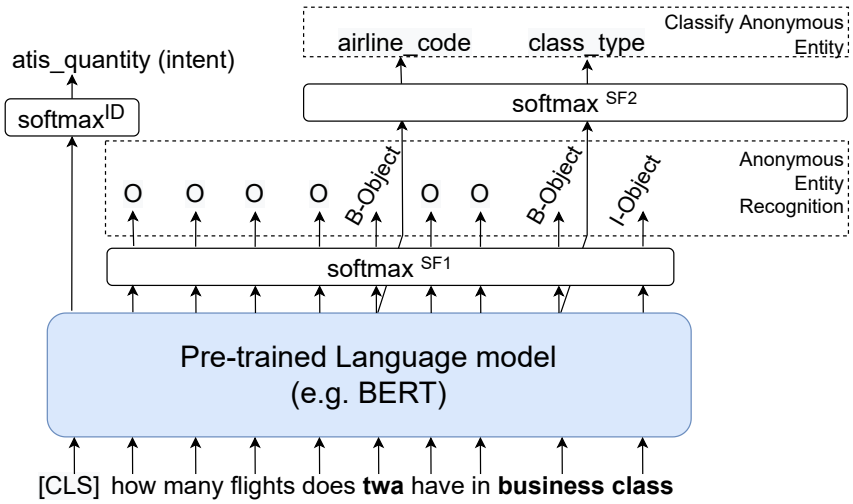


Figure 3.4: JoinBERT-CAE model architecture using joint ID and SF sub-tasks incorporating our proposed mechanism (CAE).

Conditional Random Field.

Many previous works show the use of CRF layer incorporating with neural network on the top of model architecture to support sequence label tagging Lafferty et al. [2001], Ma and Hovy [2016], Chen et al. [2019]. Therefore, it is potential to adapt this architecture to our proposed model. We aim to utilize the strong relation between Intent and Slot types, so our CRF layer is constructed to process Intent and the Slots (or entities) without considering outside entity words. In detail, we treated Intent as a Slot class of special token *[CLS]*. Difference from Chen et al. [2019], Dao et al. [2021], CRF layer in these works only considers relations between Slot types without Intent information.

$$\text{score}(\mathbf{s}, \mathbf{y}) = \sum_{i=1}^{|s|}(\mathbf{W}^e \mathbf{h}_i^{wordEntity} + \mathbf{b}^e)[\mathbf{y}_i] \; + \sum_{i=0}^{|s|}(\mathbf{W}^t[\mathbf{y}_i, \mathbf{y}_{i+1}]) \tag{3.9}$$

$$p(\mathbf{y}|\mathbf{s}) = \frac{\exp(\text{score}(\mathbf{s}, \mathbf{y}))}{\sum_{\mathbf{y}'} \exp(\text{score}(\mathbf{s}, \mathbf{y}'))} \tag{3.10}$$

where $\mathbf{y}_0, \mathbf{y}_{|s|+1}$ is additional start and end of Slot label; $\mathbf{W}^e, \mathbf{b}^e, \mathbf{W}^t$ are the learnable parameters for emission and transmission scores. By using CRF layer, the model is trained to maximize the log-probability of gold Slot sequence labels.

Local Context Integration.

We aim to evaluate the contribution of local context to SLU tasks when using a pre-trained language model, therefore, we study the effective way to inject phrase information into it. Compared with the training process of PhraseTransformer architecture trained from scratch, in this model, weights of Encoder are initialized from a pre-trained model. For mitigating the catastrophic forgetting of previous knowledge, we only apply the local context extraction layer on the top of the text Encoder component (e.g. BERT).

We use Phrase function (Equation 2.7) to extract local context features. Let define $n\_\text{gram}_k(\mathbf{s}) = [\mathbf{h}_{k-n+1}^{word}, \mathbf{h}_{k-n+2}^{word}, ..., \mathbf{h}_k^{word}]$ is the procedure extracting n neighbor hidden states of word index k given a sentence ($\mathbf{s}$) (similar Equation 2.9). After that, we integrate phrase features into word vector representation for context fusing. The main idea is to

build up the attention scores of words with neighbor words and accumulate them to get a new representations of words. Then, the original and new word features are concatenated to achieve the final representation ($\mathbf{h}'_k$):

$$\mathbf{c} = \text{Phrase}(\{\mathbf{h}_k^{word}\}_{k=1}^{|s|}, n) \tag{3.11}$$

$$\mathbf{l}_k = n\_\text{gram}_k(\mathbf{c}) \tag{3.12}$$

$$\mathbf{h}_k^{lo} = \text{softmax}(\frac{\mathbf{W}^q\mathbf{h}_k^{word}(\mathbf{W}^k\mathbf{l}_k)^\intercal}{\sqrt{d_h}}) \cdot \mathbf{W}^v\mathbf{h}_k^{word} \tag{3.13}$$

$$\mathbf{h}'_k = [\mathbf{h}_k^{word}; \mathbf{c}_k; \mathbf{h}_k^{lo}] \tag{3.14}$$

where n is the hyper-parameter gram size or local context length, $|s|$ is sentence length, d_h is hidden size of word vector, $\mathbf{W}^*$ is the learnable weights, $[\cdot;\cdot]$ is the concatenation method. Finally, the new hidden state of word index k ($\mathbf{h}'_k$) is replaced for the original word hidden state ($\mathbf{h}_k^{word}$) in Equation 3.4.

3.4 Experiments and Analysis

3.4.1 Datasets

We conducted experiments on three public benchmark SLU datasets, Snips [Coucke et al., 2018], ATIS [Hemphill et al., 1990], Vietnamese ATIS [Dao et al., 2021]. Besides, to prove the generalization of our proposed model, we also verify performance of our model on CoNLL 2003 dataset for NER task [Tjong Kim Sang and De Meulder, 2003]. Snips dataset contains 13,084 training samples, 700 testing samples, and 700 development samples. English ATIS and Vietnamese ATIS datasets are the same sizes with 4,478 training samples, 893 testing samples, and 500 development samples. With the Vietnamese ATIS dataset, we use a *word* version having data is segmented [Hemphill et al., 1990]. CoNLL2003 dataset contains 14,041 training samples, 3,453 testing samples, and 3,250 development samples. Based on our analysis (Section 3.1), the SLU datasets have more entity (Slot) classes than NER datasets, and the imbalanced class problem is more critical.

3.4.2 Experimental Settings

We aim to evaluate the performance of our proposed model, we organized experiments using JointBERT-CAE, JointBERT-CAE using CRF layer on top, and baseline JointBERT (re-implemented) models on all datasets. Similar to previous work [Chen et al., 2019] on ATIS and Snips datasets, with pre-trained model, we used the BERT-based setting[3] [Devlin et al., 2019a] with 12 Encoder layers, 12 heads, 768 hidden size. On ATIS Vietnamese dataset, we also used the pre-trained Vietnamese model PhoBERT[4] [Nguyen and Nguyen, 2020] with base setting. On CoNLL 2003 dataset, we used pre-trained BERT-based setting version case sensitive[5] and pre-trained RoBERTa-Large [Liu et al., 2019] model[6]. For fine-tuning hyper-parameters process, all experiments are conducted on the dev set of each dataset wich the number of epochs is selected in $\{5, 10, 20, 30\}$, the weight of SF loss (λ) is selected in $\{0.2, 0.3, 0.4, 1.0\}$, init learning rate is selected in $\{2e^{-5}, 5e^{-5}\}$. Besides, to compare with the previous approach using Oversampled training data [Grancharova et al., 2020] for imbalance entity class problem, we conducted experiments on ATIS dataset by duplicating samples minority class with ratio threshold (r) is selected in $\{0.01, 0.02, 0.03, 0.04, 0.05\}$. In these experiments, only training data is re-sampled while the test and dev set is original data. For comparison with the previous works, we use three common metrics to evaluate our experiments: Intent accuracy, Slot F1 score (entity level), and Sentence Frame accuracy.

3.4.3 Experimental Results

Main Results

We show the experimental results of our proposed models on three SLU datasets in Tables 3.1, 3.2, and NER CoNLL 2003 dataset in Table 3.3. To overcome the limits related to the experimental environments and libraries, we re-implemented the JointBERT as

[3]Downloaded from https://huggingface.co/bert-base-uncased
[4]Downloaded from https://huggingface.co/vinai/phobert-base
[5]Downloaded from https://huggingface.co/bert-base-cased
[6]Downloaded from https://huggingface.co/roberta-large

our baseline system. Firstly, we found that the Slot F1 scores of our proposed model JointBERT-CAE on three datasets achieve SOTA performance. Through comparison with our baseline, it increases 0.4 points on Snips, 0.3 points on ATIS, and 0.6 points on Vietnamese ATIS. These results proved that our proposed model using the CAE mechanism works effectively while the model size almost does not change. Therefore, it boosts the performance of the overall system in Sentence Frame Accuracy score, especially on ATIS, and Vietnamese ATIS datasets. In the comparison with the previous works on the Sentence accuracy metric, the JointBERT-CAE model improves 0.3 points on ATIS, 1.6 points on ATIS Vietnamese, and promising results on Snips. By using the CRF layer on the top of the pre-trained model, the results are slightly decreased, which is similar to the result shown in previous works [Chen et al., 2019, Dao et al., 2021]. We argue that joint learning using the pre-trained model is powerful enough in learning the relation between Intent and Slot information. Therefore, the CRF layer does not show a clear improvement.

Table 3.1: Result of our proposed models on the test set of two SLU datasets: Snips and ATIS . The bottom part of the table presents the results of experiments conducted in this work.

Model	**Snips**			**ATIS**		
	Intent	**Slot**	**Sent**	**Intent**	**Slot**	**Sent**
BERT-Joint [Castellucci et al., 2019a]	**99.0**	96.2	91.6	**97.8**	95.7	88.2
JointBERT [Chen et al., 2019]	98.6	**97.0**	92.8	97.5	96.1	88.2
Stack-propagation [Qin et al., 2019]	99.0	**97.0**	**92.9**	97.5	96.1	88.6
JointBERT (ours)	98.6	96.6	92.0	97.4	95.8	87.6
JointBERT-CAE	98.3	**97.0**	92.6	97.2	**96.3**	**88.9**
JointBERT-CAE +CRF	98.3	96.9	92.7	97.5	96.0	88.4
JointBERT-CAE +phrase	98.4	96.9	92.4	**97.8**	96.0	88.2

Local Context integration

We also conducted experiments for integrating local context by phrase features into a pre-trained BERT model using the CAE mechanism. The Slot F1 scores on the dev set of

Table 3.2: Result of our proposed models on the test set of ATIS Vietnamese dataset. The bottom part of the table presents the results of experiments conducted in this work.

Model	**ATIS (vietnamese)**		
	Intent	**Slot**	**Sent**
JointIDSF [Dao et al., 2021]	97.6	95.0	86.3
JointBERT (ours)	97.7	94.9	86.5
JointBERT-CAE	97.7	**95.5**	**87.9**
JointBERT-CAE +CRF	**97.8**	95.4	87.6
JointBERT-CAE +phrase	97.4	95.4	87.0

ATIS are depicted in Figure 3.5. This result showed that although the phrase mechanism slightly improves JointBERT using the CAE mechanism, this mechanism helps the pre-trained model to converge earlier. The results on the test set of three SLU datasets (Tables 3.1, 3.2) also show the competitive performance when integrating the phrase mechanism.

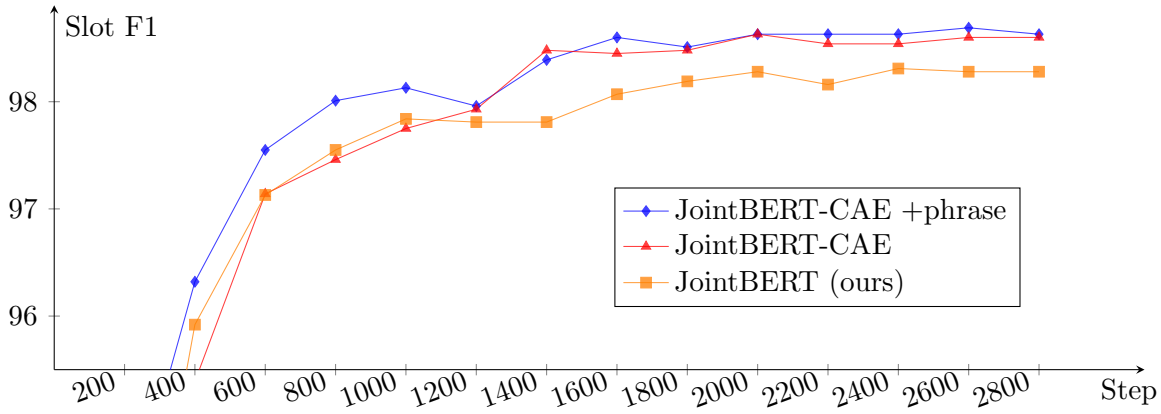


Figure 3.5: Slot F1 scores comparison between JointBERT-CAE model using phrase mechanism with baseline model on dev set of ATIS dataset.

Sequence Labeling task

Besides, we conducted extensive experiments in the NER task to inspect the generalize of our proposed model. We show the results of BERT architecture using our proposed mechanism, CAE, on dev set (Figure 3.6) and test set (Table 3.3) of CoNLL 2003 dataset. In these experiments, we constructed BERT-CAE architecture by removing the compo-

nents to learn Intent information in JointBERT-CAE architecture (Equation 3.3), and the loss of intent detection in Equation 3.8. The results on the dev set show that our CAE mechanism clearly improves the original BERT in the same setting and works effectively when incorporated with the CRF layer. The results on test set also show that our proposed mechanism improves the performance of baseline system BERT with 0.6 F1 scores, and 0.8 F1 scores when incorporating with CRF layer. Compared with the public result on this dataset, although we used the same setting described in [Devlin et al., 2019a], our baseline is lower, we argue that the reason relates to the pre-processing data and experimental libraries. Besides, we also conducted experiments using a pre-trained model RoBERTa-Large [Liu et al., 2019] for this task (Table 3.3). By using this pre-trained model, our proposed mechanism CAE increases 0.2 F1 scores when compared with the baseline model using RoBERTa large and boosts 0.3 F1 scores when incorporating with CRF layer. These results proved the solid improvement of our CAE mechanism on the different pre-trained models.

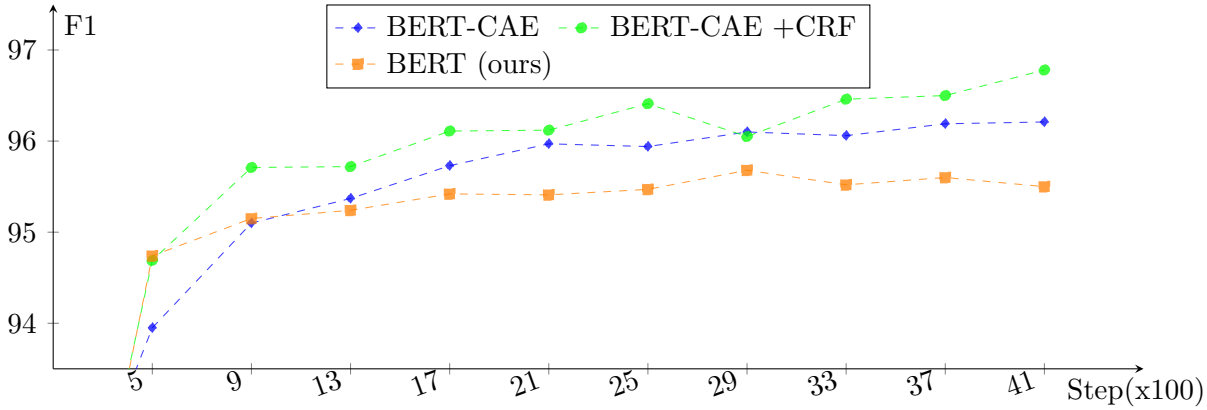


Figure 3.6: Performance comparison between our proposed mechanism (CAE) using BERT-Base model and baseline models (ours re-implemented) on dev set of CoNLL 2003 data.

Oversampled data

To evaluate our proposed approach with the previous approach [Grancharova et al., 2020] relating to the Imbalanced Entity Class problem, we re-implemented the method using oversampling training data for comparison. Figure 3.7 shows the Sentence Frame Accu-

Table 3.3: Performance comparison of the baseline models with our proposed model on the CoNLL 2003 test set.

Model	**F1 score**
BERT-Base [Devlin et al., 2019a]	92.4
BERT-Large [Devlin et al., 2019a]	92.8
BERT-Base (ours)	91.4
BERT-Base-CAE+CRF	92.2
BERT-Base-CAE	92.0
RoBERTa-Large (ours)	92.6
RoBERTa-Large-CAE+CRF	**92.9**
RoBERTa-Large-CAE	92.8

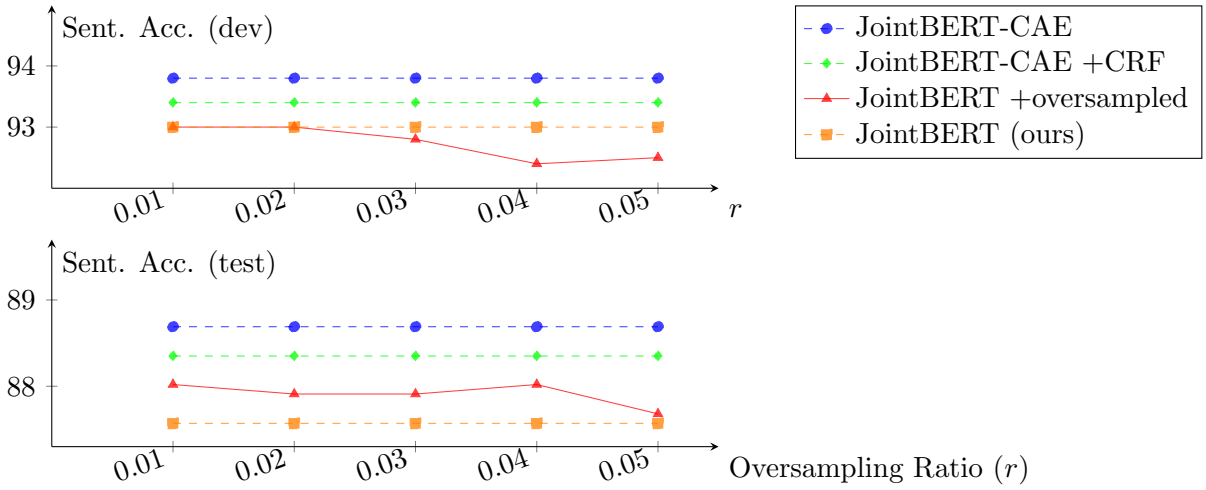


Figure 3.7: Performance comparison on the dev set (above) and test set (bellow) of ATIS dataset, among our proposed models (BERT-CAE), baseline model (JointBERT), and baseline model using Oversampling data with respect to oversampling ratio threshold.

racy of our proposed models compared with the baseline model (JointBERT) trained on oversampling training data with different ratio thresholds (r) on both dev set and test set of ATIS dataset. These results show that the re-sampling method can improve the baseline model in the small margin, however, the improvement is not solid, especially on the dev set. By using threshold $r = 0.01$, the rate of duplicated samples is 18.8%, and increase to 54.3% with $r = 0.02$. Therefore, the distribution in original data is hugely different when applying this approach, especially in strong imbalanced data like ATIS. Meanwhile, our proposed models using the CAE mechanism do not increase the size of the training dataset and still beat the re-sampling method.

3.4.4 Analysis

We conducted statistical data analysis to inspect the F1 improvement among Slot classes of our proposed model JointBERT-CAE compared with baseline model JointBERT, as shown in Figure 3.8. We found that the advancement of the JointBERT-CAE model is shown in both minority and majority classes. These results proved the generalize of the CAE mechanism. Especially, on the Snips dataset, the baseline model is typically inaccuracy in minority classes, therefore, the JointBERT-CAE showed a strong advance in these classes.

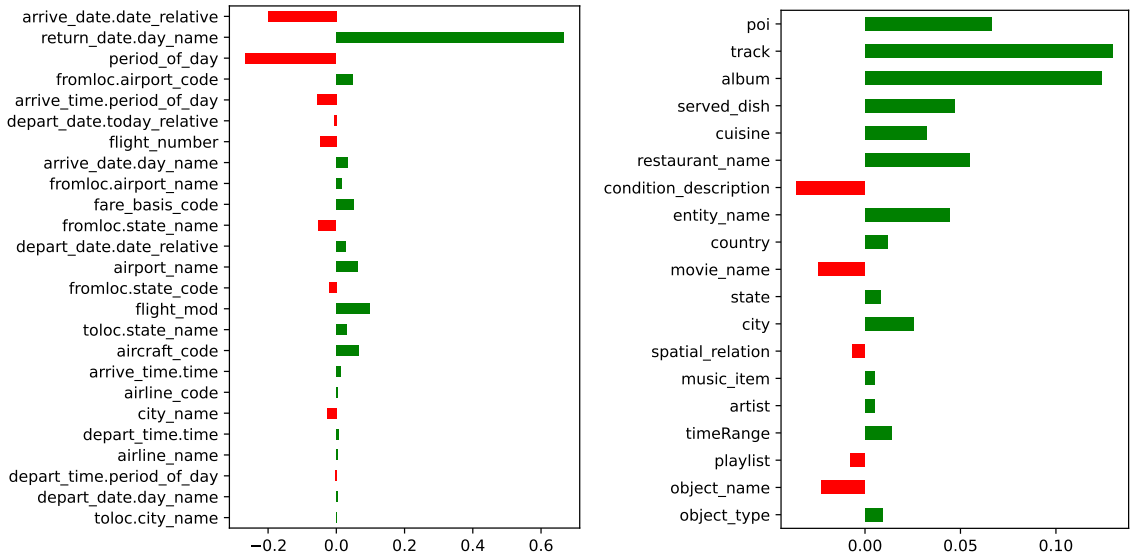


Figure 3.8: Distribution of Slot F1 improvement between JointBERT-CAE comparing with JointBERT in the test set of ATIS (left) and Snips (right) datasets. The order of Slot classes is sorted from minority to majority class. Green bars show the improvements of the JointBERT-CAE model, and the decreases are shown in red bars.

3.5 Conclusion

In this paper, we introduced a novel architecture JointBERT-CAE that work effectively on SLU datasets having highly imbalanced Slot class problem. We conducted the experiments and showed solid improvements on three SLU datasets Snips, ATIS, and Vietnamese ATIS as well as on a NER dataset CoNLL 2003. Especially, the performance of our model leads the SOTA result on ATIS Vietnamese dataset with a substantial margin compared with

previous works. Besides, the analyses statistical the output data on two well-known datasets Snips and ATIS confirmed the generalization of our model. These results also proved that the CAE mechanism is the potential to apply to sequence labeling tasks in NLP (e.g. POS task). In future works, we would like to incorporate the CAE mechanism into various Neural Network architectures to increase the performance of the SLU model as well as the models using the Sequence Labeling technique. We believe that our proposed model can be widely applied in the NLP community and real-world applications.

Chapter 4

Semantic Parsing in the Legal Domain

General Data Protection Regulation (GDPR) is an important framework for data protection that applies to all European Union countries. Recently, DAPRECO knowledge base (KB) which is a repository of if-then rules written in LegalRuleML as a formal logic representation of GDPR has been introduced to assist compliance checking. DAPRECO KB is, however, constructed manually and the current version does not cover all the articles in GDPR. Looking for an automated method, we present our machine translation approach to obtain a semantic parser translating the regulations in GDPR to their logic representation on DAPRECO KB. We also propose a new version of GDPR Semantic Parsing data by splitting each complex regulation into simple subparagraph-like units and re-annotating them based on published data from DAPRECO project. Besides, to improve the performance of our semantic parser, we propose two mechanisms: *Sub-expression intersection* and *PRESEG*. The former deals with the problem of duplicate sub-expressions while the latter distills knowledge from pre-trained language model BERT. Using these mechanisms, our semantic parser obtained a performance of 60.49% F1 in sub-expression level, which outperforms the baseline model by 5.68%.

4.1 Introduction

General Data Protection Regulation[1] is the regulation on the protection of EU citizens regarding the processing of personal data on the free flow within the European Union and on the transfer to third countries and international organizations. GDPR introduces a number of obligations that public administrations, enterprises and non-profit organizations need to observe when processing personal data. Because manual legal compliance checking is a time-consuming task, there has been an increasing interest in research on legal reasoning tools to automate the check.

When GDPR was first issued, there is a lack of logic representation for this document that can suffice to automate legal reasoning. Filling that lack, the DAPRECO knowledge base [Robaldo et al., 2020] which is a repository of if-then rules representing the regulations in GDPR has been introduced. DAPRECO KB uses the Privacy Ontology (PrOnto) [Palmirani et al., 2018] which models legal concepts in GDPR and also provides additional concepts which are needed to represent the semantics of the legal rules in GDPR. Following the Input/Output framework for legal reasoning [Sun and van der Torre, 2014], an ordinary legal rule in DAPRECO KB is usually represented by one constitutive norm (Entailment) and one regulative norm (Obligation or Permission) while a complex rule may have more norms of one or both types. The current version of DAPRECO KB is constructed manually and does not cover all articles in GDPR. This paper presents a machine translation approach to build a semantic parser which can automatically convert the regulations in GDPR to their corresponding logic representation on DAPRECO KB.

The challenge of constructing a semantic parser for logic representation on DAPRECO KB comes from its constraints in the legal domain. For example, for reasoning, the explicit representation of time in logic terms is a mandatory requirement in almost legal terms [Robaldo et al., 2020]. Therefore, there are a majority of sub-expressions are duplicated in DAPRECO KB logic for representing the conditions in GDPR statements.

It is difficult to map directly a complex GDPR rule into its original logic expression

[1] https://gdpr-info.eu/

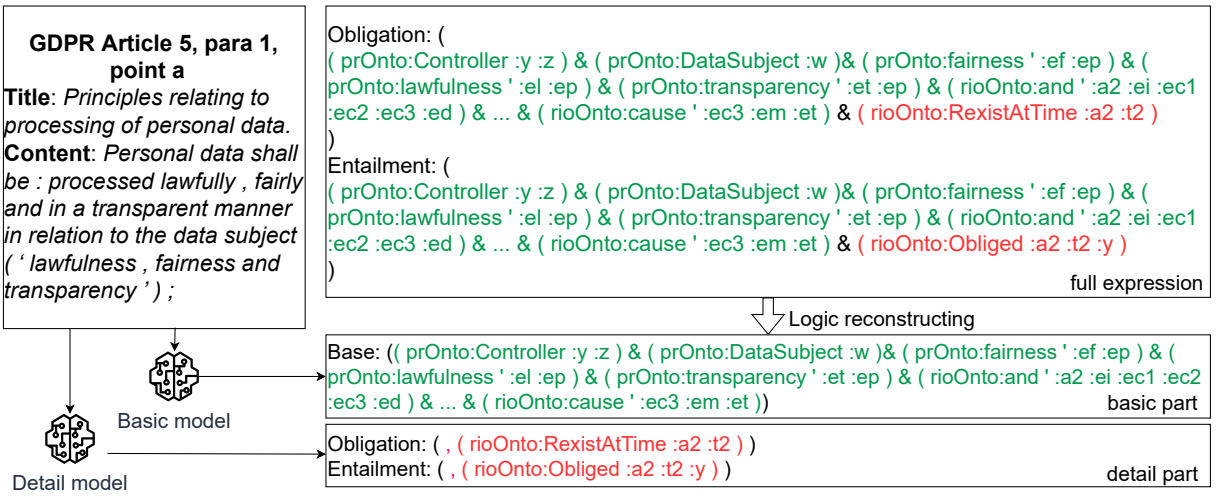


Figure 4.1: Overview of Logic mapping GDPR on DAPRECO KB using sub-expression intersection mechanism.

consisting of multiple logic formulae. To approach the challenging task, we split a complex GDPR statement into simple legal rules and then build a model to generate logical representation of these simpler rules, inspired by the research on Semantic Parsing and Question Answering dealing with complex sentences [Min et al., 2019, Zhang et al., 2019]. As a result, we constructed two versions of the GDRP Semantic Parsing dataset. The first version of the dataset (**Original data**) consisting of 275 samples is constructed from the current version of the DAPRECO KB. One sample is a pair of a GDPR statement and its logic expression. For example, the logic representation of Article 5, paragraph 1, point a is a full expression (Figure 4.1) consisting of 2 logic formulae (Obligation and Entailment). Each logic formula is an if-then rule which is a combination of sub-expressions (e.g., `(prOnto:Controller :y :z)` is a sub-expression). Similar to DAPRECO KB, a complex GDPR statement in the Original data is represented by more than two logic formulae and the number of these formulae can vary. To assist in solving the task of mapping a complex GDPR rule into its logic expression, we constructed a second version of dataset called **Relaxation data**. In this version, a complex sample is split into simple subparagraph-like units.

Concerning our semantic parser, we use machine translation approach and propose two mechanisms to improve the performance. Based on our observation on the GDPR expres-

sions, for one GDPR statement there are a lot of duplicate sub-logic expressions in its logic formulae. To avoid them, we propose Sub-expression intersection mechanism. We separate shared sub-logic expressions (basic part) from the remaining (detail part) collected from the logic formulae of GDPR statements and use them to build two Transformer-based Neural Machine Translation (NMT) models [Vaswani et al., 2017a] to generate the basic part and the detail part, respectively. For the basic part, we propose PRESEG (*i.e.*, **P**redicate **RE**trieval & **S**ub-**E**xpression **G**eneration) mechanism which consists of two steps. First, we utilize the power of the pre-trained language model BERT [Devlin et al., 2019a] to retrieve well-relevant predicates. After that, we apply a Transformer-based NMT model to generate sub-expressions for each predicate instead of generating the logic representation for the whole GDPR statement, which results in a more correct syntax of logic representation.

We evaluate our model on two versions of the dataset as mentioned above. We performed five experiment runs as the progress of developing our semantic parser. The proposed model achieves a performance of 60.49% F1 in the sub-expression level, which outperforms its baseline model.

4.2 Related Work

GDPR text extension (GDPRtEXT) [Pandit et al., 2018] provides a hierarchy of concepts present in the GDPR. For example, identified data types such as personal data and anonymous data are defined as sub-classes of the common term Data. However, the GDPRtEXT does not really model the norms and legal axioms (e.g., the actions performed by the processor, the obligations of the controller and the rights of the data subject). Moreover, GDPRtEXT does not foster FRBR information for managing the versioning of the legal text over time and consequently the changes of the legal concepts due to modifications in the legal system.

GDPRov Pandit and Lewis [2017] is an OWL2 ontology for describing the provenance of data and consent life-cycles in the light of the linked open data principles such as fairness

and trust. It extends the existing linked open data provenance ontologies - PROV ontology and ontology for Provenance and Plans. GDPRvo uses these provenance ontologies to express a data-flow model that can trace how consent and data are used by using GDPR terminology.

ODRL[2] provides predicates and classes for managing obligations, permissions, prohibitions, but several parts of the deontic logic are missing (e.g., right and penalty classes). ODRL is good for modeling simple policies, but it is quite limited to manage the complex organization of the legal rules (e.g., exceptions in the constitutive rules or in prescriptive rules).

Privacy Ontology (PrOnto) [Palmirani et al., 2018] is designed in such a way that it models the essential legal concepts in the GDPR. PrOnto has been developed following thorough ontology development methodology called MeLON. PrOnto reuses existing ontologies: ALLOT, FRBR, LKIF, the Publishing Workflow Ontology, Time-indexed Value in Context and Time Interval. However, using the PrOnto alone only allows for basic reasoning, it is not sufficient to assess compliance checking.

In 2020, Robaldo et al. [2020] introduced the DAPRECO knowledge base, which is a repository of rules codified in LegalRuleML [Palmirani et al., 2011]. The rules represent the provisions of the GDPR. The DAPRECO knowledge base was built upon the PrOnto and added additional constraints in the form of if-then rules formalized in reified Input/Output logic [Robaldo and Sun, 2017]. To date, the DAPRECO knowledge base is the biggest knowledge base in LegalRuleMl, which allows complicated legal reasoning and suffice to check compliance. The DAPRECO knowledge base is used in this work for this particular reason.

4.3 Methodology

To deal with the task of mapping a GDPR statement into its logic representation on DAPRECO KB, we apply the solution of the Semantic Parsing task in Natural Language

[2]https://www.w3.org/ns/odrl/

Processing (NLP) [Wang et al., 2015, Dong and Lapata, 2018, Jia and Liang, 2016, Wang et al., 2020, Chen et al., 2019]. With the approach using Intent Classification and Slot Filling [Wang et al., 2020, Chen et al., 2019], each logic representation is considered a semantic frame with the defined set of intent and slot information. This method requires annotated data to contain the label of slot information and intent type for each sample, which is difficult to extract from the GDPR data. A more flexible approach is using Neural Machine Translation (NMT) [Dong and Lapata, 2018, Jia and Liang, 2016]. By considering source sentences and logic representations as source and target languages in the machine translation system, the semantic parser can be adapted to any logic representation syntax. Using this method, we build a semantic parser that can map GDPR statements into their logic representations on DAPRECO KB. Besides, we propose *PRESEG* mechanism which incorporates a pre-trained language model (e.g. BERT) and a NMT model to utilize the advantages of both: *knowledge distillation* capacity and *flexible generation*.

4.3.1 Baseline NMT Model

We use Transformer architecture [Vaswani et al., 2017a] as our strong baseline model because this model is shown to be effective in learning long-range dependency, which is appropriate for a long document. In this architecture, the input is a sequence of words in a GDPR statement ($\mathbf{x} = [x_1, x_2, ..., x_{|S|}]$ where $|S|$ is the sentence length). This input is embedded by an Embedding Layer, and feed-forward via $N\times$ Transformer Encoder stacked layers to get the final vector representation. After that, the Transformer decoder based on the attention mechanism is used to decode each token in the expression ($\mathbf{y} = [y_1, y_2, ..., y_{|E|}]$ where $|E|$ is the number of tokens in the expression).

4.3.2 Sub-expression Intersection mechanism

Based on our observation on the GDPR expressions, for one GDPR statement there are a lot of duplicate sub-expressions in its logic formulae (sub-formulae). For example,

the GDPR expression in Table 4.1, all sub-formulae *f1* to *f4* share the sub-expression `( prOnto:DataSubject :w )`. To avoid them, we propose *Sub-expression Intersection* mechanism to split a GPDR expression into 2 parts: the basic part contains common sub-expressions among logic formulae and the detail part contains the remaining sub-expressions. By using this mechanism, the GDPR expression is shortened but still preserves all semantic information.

4.3.3 PRESEG mechanism

In this mechanism, we utilize the power of the pre-trained language model BERT [Devlin et al., 2019a] to support the expression parsing process. The parsing process is split into two steps (Figure 4.2):

- *Predicate Retrieval.* This step uses a BERT retrieval model to generate a set of predicates related to an input GDPR statement ($\boldsymbol{x}$). In detail, we construct a vocabulary of predicates ($\mathcal{V}^{predicate} = \{p_i\}$) from the training data then fine-tune the pre-trained BERT model to predict the relation between text input and each predicate $\langle \boldsymbol{x}, p_i \rangle$.

- *Sub-expression Generation.* With each predicate generated from the previous step, we concatenate it with the GDPR statement to generate corresponding sub-expressions using the NMT model. After that, all generated sub-expressions are combined to present the final expression.

Predicate Retrieval.

Inspired by Nguyen et al. [2022], given the predicate vocabulary ($\mathcal{V}^{predicate} = \{p_i\}$), we retrieve all predicates that are relevant to an input GDPR statement ($\mathbf{x}$). Following the task next-sentence-prediction (NSP) [Devlin et al., 2019a], for each pair $\langle \boldsymbol{x}, p_i \rangle$, we generate an input string by a template *"[CLS] sentence1 [SEP] sentence2 [SEP]"* where *sentence1*, *sentence2* are the predicate (p_i) and the GDPR statement ($\boldsymbol{x}$), respectively. We

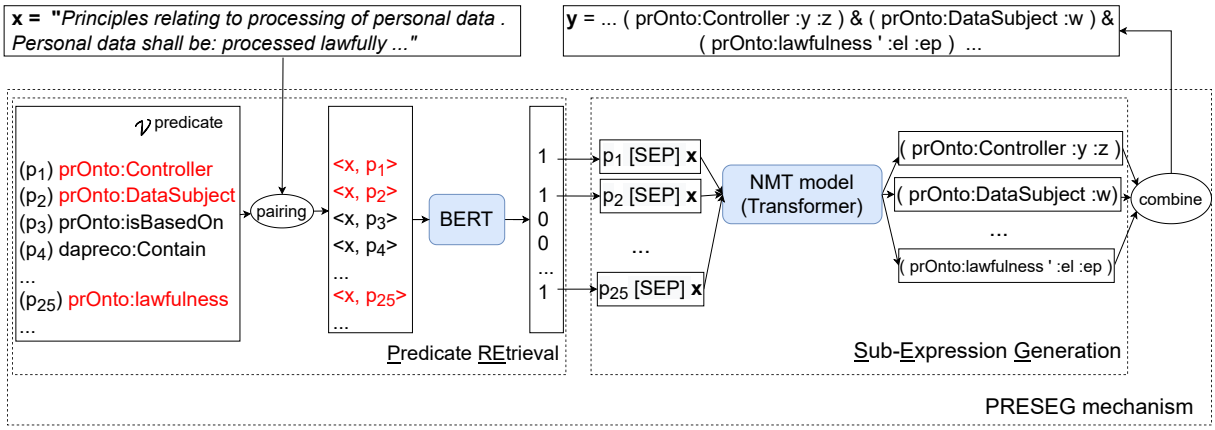


Figure 4.2: PRESEG mechanism on GDPR Article 5, para 1, point a.

use the pre-trained BERT embedding to obtain the representation for this input. Next, we forward the hidden representation of *[CLS]* token ($\boldsymbol{h}^{[CLS]}$) to a Linear layer. Finally, we use a softmax function to calculate the probability of how relevant the predicate is to the GDPR statement. Mathematically, we use the formulae as follows:

$$\boldsymbol{h}^{[CLS]}, \boldsymbol{h}^{others} = \text{BERT}(\langle p_i, \boldsymbol{x} \rangle) \tag{4.1}$$

$$\boldsymbol{h}^{out} = \boldsymbol{W}^{out}\boldsymbol{h}^{[CLS]} + \boldsymbol{b}^{out} \tag{4.2}$$

$$P(\langle p_i, \boldsymbol{x} \rangle) \propto \text{softmax}(\boldsymbol{h}^{out}) \tag{4.3}$$

where $\boldsymbol{W}^{out}$, $\boldsymbol{b}^{out}$ are learnable parameters. The loss function is *Cross-Entropy*.

Sub-expression Generation.

In this step, we construct an NMT model to generate sub-expressions for each predicate generated from the previous step based on the GDPR statement. For example (Figure 4.2), the text input ($\boldsymbol{x}$ - the GDPR Article 5, para 1, point a) and the generated predicate (p_1 = `prOnto:Controller`) are concatenated by *"[SEP]"* token. The NMT model generates the corresponding sub-expression `(prOnto:Controller :y :z)`. The architecture of the NMT model in this step is the same as the baseline NMT model based on Transformer's architecture. Compared with the baseline model, the NMT model in this step is trained to generate sub-expression instead of the full expression. In this way,

we got two advantages: the number of generated samples is 10 times larger than that of the baseline model, and the syntax of output expression is more correct. Finally, we unite all sub-expression to recover the full expression.

4.4 Experiments

In this section, we describe the detailed process to construct the GDPR Semantic Parsing dataset on DAPRECO KB and the experiments conducted on this data.

4.4.1 Datasets

We created two versions of the GDPR Semantic Parsing dataset: *Original data* - this version contains pairs of the GDPR statement and its expression (logic formulae) from [Robaldo et al., 2020]; *Relaxation data* - in this version, we split and re-annotate the complex samples to improve the consistency in all samples. By using the Relaxation data, although the semantic parser misses the target automatically mapping the whole GDPR statement into its expression, it has a meaning in verifying the improvement capacity of decomposing complex GDPR statement approach for the futures works or building a suggestion system for logic annotators on DAPRECO KB with higher accuracy.

Original data.

To construct the dataset, we crawl the content of the articles (GDPR statements) from General Data Protection Regulation homepage and the logic representations of the regulation in GDPR from the DAPRECO repository[3] [Robaldo et al., 2020]. Then we map each structural item in the GDPR article including the paragraph, the point in the paragraph to the corresponding logic formulae in the DAPRECO knowledge base. The mapping process is shown in Table 4.1. In detail, the mapping process between the GDPR statement and its expression is based on metadata information. For example, the paragraph 3 of

[3]https://github.com/dapreco/daprecokb/blob/master/gdpr/rioKB_GDPR.xml

article 37 is mapped to `refID="GDPR:art_37__para_3"` in published DAPRECO repository. Although the logic representations in fact for DAPRECO KB are written in XML syntax [Robaldo et al., 2020], we use the text version of these expressions because they have exactly the same semantic meaning and can be trivially converted back and forth. The final dataset has 275 samples: 198 normal samples and 97 complex samples.

Table 4.1: GDPR Mapping Example. This table is split into 2 parts, the upper part contains metadata information of each GDPR statement and its corresponding expression, the lower part shows their contents.

GDPR statement	**GDPR expression**
Article: 37 **Paragraph**: 3 **Sub-para**: None **Point**: None	<lrml:LegalReference refersTo="gdprC4S4A37P3-ref" refID="GDPR:art_37__para_3" />
Title: Art. 37 GDPR Designation of the data protection officer **Content**: Where the controller or the processor is a public authority or body, a single data protection officer may be designated for several such authorities or bodies, taking account of their organisational structure and size.	*(f1)* Permission: (... (prOnto:DataSubject :w) & (prOnto:Controller :y2 :z) , (rioOnto:RexistAtTime :a2 :t2) ...) *(f2)* Entailment: (... (prOnto:DataSubject :w) & (prOnto:Controller :y2 :z) , (rioOnto:Permitted :a2 :t2 :w) ...) *(f3)* Permission: (... (prOnto:DataSubject :w) & (prOnto:Processor :x1) , (rioOnto:RexistAtTime :a2 :t2) ...) *(f4)* Entailment: (... (prOnto:DataSubject :w) & (prOnto:Processor :x1) , (rioOnto:Permitted :a2 :t2 :w) ...)

Relaxation data.

The original data contains 97 complex samples. A complex one is defined as having more than two logic formulae in its logic expression. For example, the GDPR expression of Article 37, paragraph 3 (Table 4.1), has 4 logic formulae (*f1* - *f4*). Another example, the GDPR expression of Article 35, paragraph 3, subparagraph 1, point b, has 20 logic formulae. Because the number of logic formulae for each GDPR statement varies, it is difficult for a semantic parser can generalize this inconsistency with limited samples like the original data. To assist in solving the task of mapping a complex GDPR rule into its logic expression, we constructed a *Relaxation* version of this data in which the complex

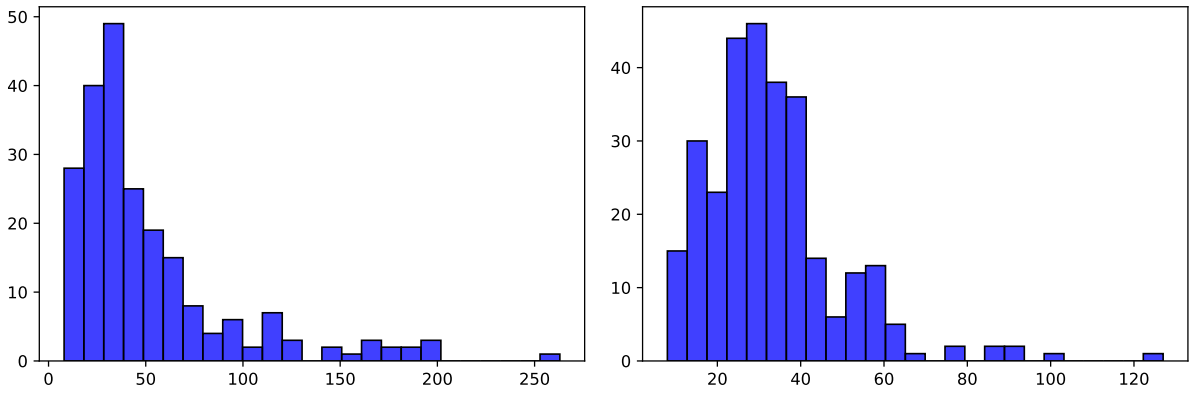


Figure 4.3: Histogram comparison of number of sub-expressions on GDPR Semantic Parsing dataset between two versions: Original (left) and Relaxation (right).

sample is split into simple subparagraph-like units and re-annotated. For example, the complex sample in Table 4.1 is split into two new samples: the former refers to the controller while the latter refers to the processor. Finally, the new dataset (relaxation version) consists of 390 samples: 198 initial ordinary samples and 192 new samples by splitting 97 complex ones. We split randomly 390 samples into training and testing datasets (the split ratio is 80:20). In Figure 4.3, we show the histogram comparison of the number of sub-expressions between Original and Relaxation versions. In the original data, there are many complex samples having numbers of sub-expressions larger than 100, while the number of sub-expressions in the relaxation version is focused in a range less than 60. Table 4.2 shows the statistic of the two versions of the datasets. The statistic shows that the GDPR semantic parsing version relaxation has 40% more samples than the original version with the shorter target expression.

Table 4.2: GDPR Semantic Parsing Data Analysis

		Original	**Relaxation**
Number of samples		275	390
GDPR statement	Average number of words	89	74
	Number of words	[17 - 1130]	[17 - 1130]
GDPR expression	Average number of tokens	401	275
	Number of tokens	[58 - 2023]	[18 - 1371]

4.4.2 Experimental Settings

For all experiments below, we set hyperparameter values according to the best setting of IWSLT 14 English-German NMT dataset [Vaswani et al., 2017a], with 3 Transformer layers (because the size of this dataset is small), hidden size is 512, 4 heads, and 200 training epochs. We conduct five experiments as follows:

- **Setting 1**: (Baseline Model) We employed a single NMT model to map the GDPR statement into its expression. In this setting, all logic formulae for each GDPR statement are concatenated to present the target GDPR expression.
- **Setting 2**: In this setting, we aim to evaluate the effectiveness of *Sub-expression intersection* mechanism. We applied this mechanism on GDPR expressions to get the basic and the detailed parts. Then we employed a single NMT model for parsing. For each GDPR statement, the target GDPR expression is the concatenation of its basic part and detail part.
- **Setting 3**: In this experiment, we aim to evaluate the effectiveness of *sub-expression intersection* by generating separately two logic parts. Instead of employing a single NMT model as experiment 2, we employed two NMT models to learn the basic part and detail part separately.
- **Setting 4**: In this experiment, we aim to evaluate the effectiveness of *PRESEG* mechanism. We used *Sub-expression intersection* mechanism similar to experiment 3. Then we employed *PRESEG* mechanism for the basic part.

4.4.3 Experimental Results and Discussion

Main result. Table 4.3 shows the performance on the Original test set and the Relaxation test set for our experiments, respectively. On the Original data, the performance of employing a single NMT model to learn full logic representation directly is average, $F1 = 37.16\%$ (setting 1). When the complex data samples were split and re-annotated (relaxation version), the performance increased by 17.39% ($F1 = 54.55\%$ in setting 1).

Similar to experiment 1, the performance of remaining experiments when learning logic representation on Relaxation data also increased compared to on Original data. These improvements show that the GDPR expressions in Relaxation data are more consistent than the original version, which makes the model more generalizable. These results show the effectiveness of decomposing the complex GDPR statement into simple ones.

Using the *Sub-expression intersection* mechanism, on the full expression of Original data, the performance in experiments 2 and 3 increased respectively by 14.25% and 3.27% compared to the baseline model (setting 1). Similar to the setting on the original version, the F1-score on Relaxation data increased by at least 3% compared to the baseline model. We argue that this mechanism filters the duplicate sub-expressions in the GDPR sub-formulae; using this mechanism can reduce the complexity of the GDPR logic representation but still preserve the original semantic information. Besides, the performance in setting 2 when using this mechanism with a single NMT model (end-to-end model) is better than using separately two NMT models for the basic and detail parts because the end-to-end model utilizes the relation between these parts to improve parsing logic representation.

Table 4.3: Result of our experiments on GDPR Semantic Parsing data. The notation "n/a" indicates that the measurement method is not applicable.

		Setting 1	**Setting 2**	**Setting 3**	**Setting 4**
Single NMT model		✓	✓		
Multi NMT models				✓	✓
+ Sub-expression Intersection			✓	✓	✓
+ PRESEG					✓
Original data	Full Expression	37.16	**51.41**	40.43	38.59
	Basic Part	n/a	**68.41**	47.20	44.73
	Detail Part	n/a	**14.69**	1.82	1.82
Relaxation data	Full Expression	54.55	57.76	57.30	**60.23**
	Basic Part	n/a	62.30	61.32	**64.22**
	Detail Part	n/a	**32.27**	31.21	31.21

Using the *PRESEG* mechanism on Relaxation data (setting 4 Table 4.3), our proposed model outperforms all previous experiments with $F1 = 64.22\%$ in the basic part (increased by 1.92% compared to experiment 2 and increased by 2.9% when compared to setting 3). It boosts the performance on full logic representation to 60.23%, F1 increased by 2.47%

compared to experiment 2 and increased by 2.93% compared to setting 3. However, this mechanism did not show improvement on the original version. The reason is that the GDPR expressions in the Original data are inconsistent.

Local context integration. In this experiment, we aim to apply the PharseTransformer architecture (PhraseTrans.$_{CrossH}$) to improve the performance of Semantic Parser. Based on the results of previous experiments, we used setting 2 which uses a Sub-expression intersection mechanism to learn the whole logic representation by one NMT model. The experimental results (Table 4.4) show that the PhraseTransformer beat the original Transformer model on both two versions GDPR semantic parsing dataset. Besides, the PhraseTransformer shows the advantage more clearly in the basic part than the detail part of logic representation, with a 1.08 F1 score in Original data and a 2.62 F1 score in Relaxation data. The basic part is commonly contain base sub-expressions that refer to general conditions in GDPR points (e.g. `prOnto:DataSubject` condition). We argue that the reason comes from the frequent phrases of base condition, which PhraseTransformer supports to represent meaning better than the vanilla Transformer.

Table 4.4: Performance comparison (F1 score) between the original Transformer and PhraseTransformer on GDPR Semantic Parsing data using single NMT model.

		Setting 2 (Transformer)	**PhraseTransformer**
Original data	Full Expression	**51.41**	**51.48**
	Basic Part	68.41	**69.49**
	Detail Part	14.69	**15.48**
Relaxation data	Full Expression	57.76	**58.72**
	Basic Part	62.30	**64.92**
	Detail Part	**32.27**	24.61

Error Analysis. An analysis of mispredicted logic representation in the test set showed three main causes of generating errors relating to variables in sub-expressions. The variable names, which are named by human annotators, are usually not meaningful names. For example, in the GDPR expression of Article 5, paragraph 1, point a, `x`, `ep` are variable names for the controller, and the predicate `PersonalDataProcessing`, respectively. With limited data, that is not easy for a model to learn the way of naming variables if

the annotators do not annotate variable names consistently. In addition, errors also occur in predicting the position of predicates. Instead of correctly predicting that a predicate belongs to the if statement, the model sometimes predicted that the predicate belongs in the then statement, and vice versa. Moreover, the model could not identify the predicates which rarely appear in logic representations.

The Need for Correct Variables in Sub-expression Component. One natural question is how often our semantic parser fail to generate the correct variable. To answer this, we conducted evaluation experiments with oracle variable information (Table 4.5). For setting using oracle variable name, for matching sub-logic expression in the evaluation process, the different variable is ignored. In other words, these results show the accuracy of predicate (function name) in output logic representation.

Table 4.5: F1 on the test set given an oracle providing correct number of variables and variable names in each sub-logic expression

	Basic Part	**Detail Part**	**Full Expression**
Setting 4	64.22	31.21	60.23
+Oracle variable	69.05 (+4.83)	69.82 (+38.61)	67.72 (+7.49)

With oracle "variable", we observed a F1 of 69.05% for the basic part, 69.82% for the detail part, and 67.72% for full expression. This verifies that if the model can learn well the constraints between variables in each sub-logic expression according to each predicate, the performance of model can increase a lot. Therefore the problem of generating correct constraint between variable and predicate requires important future work.

4.5 Conclusion

In this paper, we propose an effective semantic parser for mapping GDPR to corresponding logic representation on DAPRECO KB. Firstly, we create Relaxation data for this task by splitting and re-annotating the complex regulation. Secondly, we introduce Sub-expression intersection mechanism to solve the problem of generation of duplicate sub-logic

expressions. Last but not least, we demonstrate how PRESEG mechanism utilized the power of the pre-trained language model BERT and the Transformer-based NMT model to generate the basic part in the logic representations. Empirically, our proposed model allows us to gain significant improvement on mapping the GDPR statement to its logic representation when compared to baseline model. Our semantic parser will be beneficial in tasks such as mapping other legal rules to logic representations.

In the future, we look forward to improving the architecture design by considering the constraint between the variable and predicate in each sub-expression.

Chapter 5

Conclusion and Future Work

5.1 Conclusions

In this thesis, we study the task of Semantic Parsing in NLP, which plays a key role in building human language interfaces, or human-machine communication. The main findings and our contributions are discussed and discuss and summarized as follows:

- **Local context integration** (Chapter 2): We expose the important role of local context information by modeling phrases in a semantic parsing task. We proposed a deep learning model for the sequence generation task, the PhraseTransformer, that works effectively in capturing relations between phrases in the sentence encoding process. The experimental results show that our proposed model improved the performance of semantic parsing task on two well-known benmark datasets Geo, Atis, and achieved competitive results on MSParS. Besides, we also show the generalize of our proposed model by adapting to Machine Translation task. The PhraseTransformer model showed the solid improvement on three Machine Translation datasets (IWLST14 German-English, IWSLT15 Vietnamese-English, WMT14 English-German).

- **Class Imbalance in SLU** (Chapter 3): We reveal the strong effect of class imbalance among slots in the Spoken Language Understanding system which is a seman-

tic parser in a task-oriented dialog system. We propose the Classify Anonymous Entities mechanism by solving the Slot Filling task with two sub-tasks, detecting anonymous entities, and classifying recognized anonymous entities. The experimental results show that the proposed model promotes the performance of the semantic parsing model, principally in the minority class of Slot recognition. Besides, we also present the effective way of integrating local context into the pre-trained language model and its contribution to this task. In addition, our experiments on the NER task also show the improvement that is proof of the applicability of our CAE mechanism to other tasks using the sequence labeling approach.

- **Semantic Parsing in the Legal Domain** (Chapter 4): We aim to apply the state-of-the-art methods of semantic parsing tasks in the legal domain and show effective ways to deal with the complex constraints in this domain. We firstly re-construct the Semantic Parsing GDPR dataset based on DAPRECO KB and formulate the semantic analysis problem in this dataset. We proposed two mechanisms to build a Semantic Parser on this data, *Sub-expression intersection* and *PRESEG* to deal with the complex constraints problem. The experimental results show the strong improvement of our proposed mechanisms. Furthermore, we also conducted experiments integrating local context into the semantic analysis model and show improved results in this domain.

5.2 Future work

Based on the current results, there are some potential directions that can be further studied in the future work:

- In this study, our PhraseTransformer only focuses on modeling phrases to reinforce sentence vector representation. However, the approach of PhraseTransformer can be adapted to paragraph or document representation. Furthermore, the sentence, paragraph hidden representation can be injected into the self-attention mechanism

that complements the dependencies among sentences or paragraphs in a hierarchical way. The improvement of paragraph or document vector representation can be applied for Retrieval documents, or Document summarization tasks.

- Based on the detailed analysis in PhraseTransformer architecture experiments about hidden phrases, not only sentence vector representations are improved, but also the phrase representations are enhanced. The meaning of phrases is synthesized by constituent words (captured by LSTM) and the context of that phrase belongs (captured by the Self-attention mechanism). Therefore, it can be applied to keywords/keyphrases extraction tasks.

- Related to the CAE mechanism in the SLU task, our extensive experiments on the NER dataset ConLL 2003 show the effectiveness of this mechanism. The idea of the CAE mechanism can be considered a coarse-to-fine labeling process. Therefore, it is the potential to apply to nested entity recognization tasks.

- In the legal domain, although two proposed mechanisms *Sub-expression intersection* and *PRESEG* can improve the performance, the current performance is quite low for use on the real application. The problem of variable names is largely influential in the current system error. To solve this problem, the graph transducer approach with edge prediction is one of the promising ways of studying.

Bibliography

D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. Jan. 2015. 3rd International Conference on Learning Representations, ICLR 2015 ; Conference date: 07-05-2015 Through 09-05-2015.

J. Berant and P. Liang. Semantic parsing via paraphrasing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1415–1425, Baltimore, Maryland, June 2014. Association for Computational Linguistics.

B. Bogin, M. Gardner, and J. Berant. Global reasoning over database structures for text-to-SQL parsing. 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)*, pages 3659–3664, Hong Kong, China, Nov. 2019. Association for Computational Linguistics.

R. Cao, S. Zhu, C. Liu, J. Li, and K. Yu. Semantic parsing with dual learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 51–64, Florence, Italy, July 2019. Association for Computational Linguistics.

R. Cao, S. Zhu, C. Yang, C. Liu, R. Ma, Y. Zhao, L. Chen, and K. Yu. Unsupervised dual paraphrasing for two-stage semantic parsing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6806–6817, Online, July 2020. Association for Computational Linguistics.

G. Castellucci, V. Bellomaria, A. Favalli, and R. Romagnoli. Multi-lingual intent detection and slot filling in a joint bert-based model. *ArXiv*, abs/1907.02884, 2019a.

G. Castellucci, V. Bellomaria, A. Favalli, and R. Romagnoli. Multi-lingual intent detection and slot filling in a joint bert-based model. *CoRR*, abs/1907.02884, 2019b.

Q. Chen, Z. Zhuo, and W. Wang. Bert for joint intent classification and slot filling. *arXiv preprint arXiv:1902.10909*, 2019.

A. Church. *The Calculi of Lambda Conversion. (AM-6)*. Princeton University Press, 1941. ISBN 9780691083940.

A. Coucke, A. Saade, A. Ball, T. Bluche, A. Caulier, D. Leroy, C. Doumouro, T. Gisselbrecht, F. Caltagirone, T. Lavril, M. Primet, and J. Dureau. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *CoRR*, abs/1805.10190, 2018.

D. A. Dahl, M. Bates, M. Brown, W. Fisher, K. Hunicke-Smith, D. Pallett, C. Pao, A. Rudnicky, and E. Shriberg. Expanding the scope of the atis task: The atis-3 corpus. In *Proceedings of the Workshop on Human Language Technology*, HLT '94, page 43–48, USA, 1994. Association for Computational Linguistics. ISBN 1558603573.

M. H. Dao, T. H. Truong, and D. Q. Nguyen. Intent Detection and Slot Filling for Vietnamese. In *Proceedings of the 22nd Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2021.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. 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)*, pages 4171–4186, Minneapolis, Minnesota, June 2019a. Association for Computational Linguistics.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. 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), pages 4171–4186, Minneapolis, Minnesota, June 2019b. Association for Computational Linguistics.

L. Dong and M. Lapata. Language to logical form with neural attention. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 33–43, Berlin, Germany, Aug. 2016. Association for Computational Linguistics.

L. Dong and M. Lapata. Coarse-to-fine decoding for neural semantic parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 731–742, Melbourne, Australia, July 2018. Association for Computational Linguistics.

N. Duan. Overview of the nlpcc 2019 shared task: Open domain semantic parsing. In J. Tang, M.-Y. Kan, D. Zhao, S. Li, and H. Zan, editors, *Natural Language Processing and Chinese Computing*, pages 811–817, Cham, 2019. Springer International Publishing. ISBN 978-3-030-32236-6.

M. Ellsworth, C. Baker, and M. R. L. Petruck. FrameNet and typology. In *Proceedings of the Third Workshop on Computational Typology and Multilingual NLP*, pages 61–66, Online, June 2021. Association for Computational Linguistics.

C. J. Fillmore and C. F. Baker. Frame semantics for text understanding. In *Proceedings of WordNet and Other Lexical Resources Workshop, NAACL*, volume 6, 2001.

D. Ge, J. Li, and M. Zhu. A transformer-based semantic parser for nlpcc-2019 shared task 2. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 772–781. Springer, 2019.

X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In Y. W. Teh and M. Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings*

of Machine Learning Research, pages 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR.

O. Goldman, V. Latcinnik, E. Nave, A. Globerson, and J. Berant. Weakly supervised semantic parsing with abstract examples. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1809–1819, Melbourne, Australia, July 2018. Association for Computational Linguistics.

C.-W. Goo, G. Gao, Y.-K. Hsu, C.-L. Huo, T.-C. Chen, K.-W. Hsu, and Y.-N. Chen. Slot-gated modeling for joint slot filling and intent prediction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 753–757, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

M. Grancharova, H. Berg, and H. Dalianis. Improving named entity recognition and classification in class imbalanced swedish electronic patient records through resampling. In *Eighth Swedish Language Technology Conference (SLTC)*. Förlag Göteborgs Universitet, 2020.

J. Hao, X. Wang, S. Shi, J. Zhang, and Z. Tu. Multi-granularity self-attention for neural machine translation. 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)*, pages 887–897, Hong Kong, China, Nov. 2019. Association for Computational Linguistics.

K. He, Y. Yan, and W. Xu. From context-aware to knowledge-aware: Boosting oov tokens recognition in slot tagging with background knowledge. *Neurocomputing*, 445:267–275, 2021. ISSN 0925-2312.

C. T. Hemphill, J. J. Godfrey, and G. R. Doddington. The ATIS spoken language systems pilot corpus. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27,1990*, 1990.

G. G. Hendrix, E. D. Sacerdoti, D. Sagalowicz, and J. Slocum. Developing a natural language interface to complex data. *ACM Trans. Database Syst.*, 3(2):105–147, June 1978. ISSN 0362-5915.

J. Herzig and J. Berant. Don't paraphrase, detect! rapid and effective data collection for semantic parsing. 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)*, pages 3810–3820, Hong Kong, China, Nov. 2019. Association for Computational Linguistics.

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8): 1735–1780, Nov. 1997. ISSN 0899-7667.

R. Jia and P. Liang. Data recombination for neural semantic parsing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12–22, Berlin, Germany, Aug. 2016. Association for Computational Linguistics.

T. Kočiský, G. Melis, E. Grefenstette, C. Dyer, W. Ling, P. Blunsom, and K. M. Hermann. Semantic parsing with semi-supervised sequential autoencoders. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1078–1087, Austin, Texas, Nov. 2016. Association for Computational Linguistics.

P. Koehn. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain, July 2004. Association for Computational Linguistics.

T. Kwiatkowski, L. Zettlemoyer, S. Goldwater, and M. Steedman. Lexical generalization in CCG grammar induction for semantic parsing. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1512–1523, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics.

J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, page 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. ISBN 1558607781.

C. Li, L. Li, and J. Qi. A self-attentive model with gate mechanism for spoken language understanding. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3824–3833, Brussels, Belgium, Oct.-Nov. 2018. Association for Computational Linguistics.

X. Li, J. Feng, Y. Meng, Q. Han, F. Wu, and J. Li. A unified MRC framework for named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5849–5859, Online, July 2020a. Association for Computational Linguistics.

X. Li, X. Sun, Y. Meng, J. Liang, F. Wu, and J. Li. Dice loss for data-imbalanced NLP tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 465–476, Online, July 2020b. Association for Computational Linguistics.

Z. Li, Y. Lai, Y. Xie, Y. Feng, and D. Zhao. A sketch-based system for semantic parsing. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 748–759. Springer, 2019.

Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

T. Luong, H. Pham, and C. D. Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal, Sept. 2015. Association for Computational Linguistics.

X. Ma and E. Hovy. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074, Berlin, Germany, Aug. 2016. Association for Computational Linguistics.

S. Min, V. Zhong, L. Zettlemoyer, and H. Hajishirzi. Multi-hop reading comprehension through question decomposition and rescoring. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6097–6109, Florence, Italy, July 2019. Association for Computational Linguistics.

D. Q. Nguyen and A. T. Nguyen. PhoBERT: Pre-trained language models for Vietnamese. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1037–1042, 2020.

H.-T. Nguyen, M.-P. Nguyen, T.-H.-Y. Vuong, M.-Q. Bui, M.-C. Nguyen, T.-B. Dang, V. Tran, L.-M. Nguyen, and K. Satoh. Transformer-based approaches for legal text processing. *The Review of Socionetwork Strategies*, 16(1):135–155, Apr 2022. ISSN 1867-3236.

P. M. Nguyen, K. Than, and M. Le Nguyen. Marking mechanism in sequence-to-sequence model for mapping language to logical form. In *2019 11th International Conference on Knowledge and Systems Engineering (KSE) (KSE'19)*, Da Nang, Vietnam, Oct. 2019.

X.-P. Nguyen, S. Joty, S. Hoi, and R. Socher. Tree-structured attention with hierarchical accumulation. In *International Conference on Learning Representations*, 2020.

M. Palmer, D. Gildea, and P. Kingsbury. The proposition bank: An annotated corpus of semantic roles. *Comput. Linguist.*, 31(1):71–106, mar 2005. ISSN 0891-2017.

M. Palmirani, G. Governatori, A. Rotolo, S. Tabet, H. Boley, and A. Paschke. *LegalRuleML: XML-Based Rules and Norms*, volume 7018 of *Lecture Notes in Computer Science*, pages 298–312. Springer, 2011.

M. Palmirani, M. Martoni, A. Rossi, C. Bartolini, and L. Robaldo. -pronto: Privacy ontology for legal reasoning, 2018.

H. J. Pandit and D. Lewis. Modelling provenance for gdpr compliance using linked open data vocabularies. In *PrivOn@ISWC*, 2017.

H. J. Pandit, K. Fatema, D. O'Sullivan, and D. Lewis. Gdprtext - gdpr as a linked data resource. In *ESWC*, 2018.

M. Post. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels, Oct. 2018. Association for Computational Linguistics.

I. Provilkov, D. Emelianenko, and E. Voita. BPE-dropout: Simple and effective subword regularization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1882–1892, Online, July 2020. Association for Computational Linguistics.

L. Qin, W. Che, Y. Li, H. Wen, and T. Liu. A stack-propagation framework with token-level intent detection for spoken language understanding. 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)*, pages 2078–2087, Hong Kong, China, Nov. 2019. Association for Computational Linguistics.

L. Qin, X. Xu, W. Che, and T. Liu. AGIF: An adaptive graph-interactive framework for joint multiple intent detection and slot filling. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1807–1816, Online, Nov. 2020. Association for Computational Linguistics.

L. Qin, T. Liu, W. Che, B. Kang, S. Zhao, and T. Liu. A co-interactive transformer for joint slot filling and intent detection. *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8193–8197, 2021a.

L. Qin, T. Xie, W. Che, and T. Liu. A survey on spoken language understanding: Recent advances and new frontiers. In Z.-H. Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 4577–4584. International Joint Conferences on Artificial Intelligence Organization, 8 2021b. Survey Track.

S. Ravuri and A. Stolcke. Recurrent neural network and lstm models for lexical utterance classification. In *Proc. Interspeech*, pages 135–139. ISCA - International Speech Communication Association, September 2015.

L. Robaldo and X. Sun. Reified input/output logic: Combining input/output logic and reification to represent norms coming from existing legislation. *J. Log. Comput.*, 27(8): 2471–2503, 2017.

L. Robaldo, C. Bartolini, M. Palmirani, A. Rossi, M. Martoni, and G. Lenzini. Formalizing GDPR provisions in reified I/O logic: The DAPRECO knowledge base. *J. Log. Lang. Inf.*, 29(4):401–449, 2020.

R. Sennrich, B. Haddow, and A. Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany, Aug. 2016. Association for Computational Linguistics.

C. E. Shannon. A mathematical theory of communication. *ACM SIGMOBILE mobile computing and communications review*, 5(1):3–55, 2001.

X. Sun and L. W. N. van der Torre. Combining constitutive and regulative norms in input/output logic. In *Deontic Logic and Normative Systems - 12th International Conference, DEON 2014, Ghent, Belgium, July 12-15, 2014. Proceedings*, volume 8554 of *Lecture Notes in Computer Science*, pages 241–257. Springer, 2014.

I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Wein-

berger, editors, *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc., 2014.

H. Tang, D. Ji, and Q. Zhou. End-to-end masked graph-based crf for joint slot filling and intent detection. *Neurocomputing*, 413:348–359, 2020. ISSN 0925-2312.

E. F. Tjong Kim Sang and F. De Meulder. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147, 2003.

G. Tur and R. De Mori. Spoken language understanding: Systems for extracting semantic information from speech. *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech*, 03 2011. doi: 10.1002/9781119992691.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017a.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc., 2017b.

D. Waltz and B. Goodman. Planes: A data base question-answering system. *SIGART Bull.*, (61):24, Feb. 1977. ISSN 0163-5719.

B. Wang, R. Shin, X. Liu, O. Polozov, and M. Richardson. RAT-SQL: Relation-aware schema encoding and linking for text-to-SQL parsers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7567–7578, Online, July 2020. Association for Computational Linguistics.

X. Wang, Z. Tu, D. Xiong, and M. Zhang. Translating phrases in neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language*

Processing, pages 1421–1431, Copenhagen, Denmark, Sept. 2017. Association for Computational Linguistics.

Y. Wang, J. Berant, and P. Liang. Building a semantic parser overnight. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1332–1342, Beijing, China, July 2015. Association for Computational Linguistics.

Y. Wang, Y. Shen, and H. Jin. A bi-model based RNN semantic frame parsing model for intent detection and slot filling. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 309–314, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

Y. Wang, H.-Y. Lee, and Y.-N. Chen. Tree transformer: Integrating tree structures into self-attention. 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)*, pages 1061–1070, Hong Kong, China, Nov. 2019. Association for Computational Linguistics.

Y. Wang, H. Chu, C. Zhang, and J. Gao. Learning from language description: Low-shot named entity recognition via decomposed framework. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1618–1630, Punta Cana, Dominican Republic, Nov. 2021. Association for Computational Linguistics.

Y. W. Wong and R. Mooney. Learning for semantic parsing with statistical machine translation. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 439–446, New York City, USA, June 2006. Association for Computational Linguistics.

Y. W. Wong and R. Mooney. Learning synchronous grammars for semantic parsing with

lambda calculus. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 960–967, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

W. A. Woods. Progress in natural language understanding: An application to lunar geology. In *Proceedings of the June 4-8, 1973, National Computer Conference and Exposition*, AFIPS '73, page 441–450, New York, NY, USA, 1973. Association for Computing Machinery. ISBN 9781450379168.

W. Wu, H. Wang, T. Liu, and S. Ma. Phrase-level self-attention networks for universal sentence encoding. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3729–3738, Brussels, Belgium, Oct.-Nov. 2018. Association for Computational Linguistics.

D. Xie, D. Ji, H. Tang, and Q. Zhou. Match matrix aggregation enhanced transition-based neural network for sql parsing. *Neurocomputing*, 445:167–179, 2021. ISSN 0925-2312.

C. Xu, Q. Li, D. Zhang, J. Cui, Z. Sun, and H. Zhou. A model with length-variable attention for spoken language understanding. *Neurocomputing*, 379:197–202, 2020a. ISSN 0925-2312.

H. Xu, J. van Genabith, D. Xiong, Q. Liu, and J. Zhang. Learning source phrase representations for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 386–396, Online, July 2020b. Association for Computational Linguistics.

P. Xu and R. Sarikaya. Convolutional neural network based triangular crf for joint intent detection and slot filling. In *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 78–83, 2013. doi: 10.1109/ASRU.2013.6707709.

X. Xu, C. Liu, and D. Song. Sqlnet: Generating structured queries from natural language without reinforcement learning. *CoRR*, abs/1711.04436, 2017.

B. Yang, Z. Tu, D. F. Wong, F. Meng, L. S. Chao, and T. Zhang. Modeling localness for self-attention networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4449–4458, Brussels, Belgium, Oct.-Nov. 2018. Association for Computational Linguistics.

P. Yin, C. Zhou, J. He, and G. Neubig. StructVAE: Tree-structured latent variable models for semi-supervised semantic parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 754–765, Melbourne, Australia, July 2018. Association for Computational Linguistics.

T. Yu, C.-S. Wu, X. V. Lin, bailin wang, Y. C. Tan, X. Yang, D. Radev, richard socher, and C. Xiong. Gra{pp}a: Grammar-augmented pre-training for table semantic parsing. In *International Conference on Learning Representations*, 2021.

J. M. Zelle and R. J. Mooney. Learning to parse database queries using inductive logic programming. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence - Volume 2*, AAAI'96, page 1050–1055. AAAI Press, 1996. ISBN 026251091X.

L. Zettlemoyer and M. Collins. Online learning of relaxed CCG grammars for parsing to logical form. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 678–687, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

L. S. Zettlemoyer and M. Collins. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*, UAI'05, pages 658–666, Arlington, Virginia, United States, 2005. AUAI Press. ISBN 0-9749039-1-4.

H. Zhang, J. Cai, J. Xu, and J. Wang. Complex question decomposition for semantic parsing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4477–4486, Florence, Italy, July 2019. Association for Computational Linguistics.

K. Zhao and L. Huang. Type-driven incremental semantic parsing with polymorphism. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1416–1421, Denver, Colorado, May–June 2015. Association for Computational Linguistics.

A. Ziai. Compositional pre-training for neural semantic parsing. In *Proceedings of the 3rd International Conference on Natural Language and Speech Processing*, pages 135–141, Trento, Italy, Sept. 2019. Association for Computational Linguistics.

Publications and Awards

Submitted Journals

[1] P. M. Nguyen, Tung Le, V. Tran, and M. L. Nguyen. PhraseTransformer: An Incorporation of Local Context Information into Sequence-to-Sequence Semantic Parsing, *Applied Intelligence*, (revision in June 2022).

Journals

[2] H.-T. Nguyen, M.-P. Nguyen, T.-H.-Y. Vuong, M.-Q. Bui, M.-C. Nguyen, T.-B. Dang, V. Tran, L.-M. Nguyen, and K. Satoh. Transformer-based approaches for legal text processing. *The Review of Socionetwork Strategies*, 16(1):135–155, Apr 2022. ISSN 1867-3236.

Conference papers

[3] Phuong Nguyen, Thi-Thu-Trang Nguyen, Vu Tran, Ha-Thanh Nguyen, Le-Minh Nguyen and Ken Satoh. Learning to map the GDPR to Logic Representation on DAPRECO-KB. In *14th Asian Conference on Intelligent Information and Database Systems* (Accepted).

[4] P. M. Nguyen, Tung Le, and M. L. Nguyen. CAE: Mechanism to Diminish the Class Imbalanced in SLU Slot Filling Task. In *14th International Conference on*

Computational Collective Intelligence, (Accepted).

[5] P. M. Nguyen, K. Than and M. Le Nguyen. Marking mechanism in sequence-to-sequence model for mapping language to logical form. In *2019 11th International Conference on Knowlerdge and System Engineering (KSE) (KSE'19)*, Da Nang, Vietnam, Oct.2019.

[6] N. Phuong, L. Tung, H. Thanh-Le, D. Thai, T. Khanh, N. Kim-Anh, and N. Le-Minh. Improving neural machine translation by efficiently incorporating syntactic templates. In *Advances and Trends in Artificial Intelligence. Artificial Intelligence Practices.* Springer International Publishing, 2022.

[7] N. H. Thanh, B. M. Quan, C. Nguyen, T. Le, N. M. Phuong, D. T. Binh, V. T. H. Yen, T. Racharak, N. Le Minh, T. D. Vu, P. V. Anh, N. T. Son, H. T. Nguyen, B. Butr-indr, P. Vateekul, and P. Boonkwan. A summary of the alqac 2021 competition. In *2021 13th International Conference on Knowledge and Systems Engineering (KSE)*, pages 1–5, 2021. doi: 10.1109/KSE53942.2021.9648724.

[8] V. Tran, V.-H. Tran, P. Nguyen, C. Nguyen, K. Satoh, Y. Matsumoto, and M. Nguyen. CovRelex: A COVID-19 retrieval system with relation extraction. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 24–31, Online, Apr. 2021. Association for Computational Linguistics.

[9] K. Yuntao., N. Phuong., T. Racharak., T. Le., and N. Minh. An effective method to answer multi-hop questions by single-hop qa system. In *Proceedings of the 14th International Conference on Agents and Artificial Intelligence - Volume 2: ICAART,*, pages 244–253. INSTICC, SciTePress, 2022. ISBN 978-989-758-547-0.

Awards

- Ranked second place among all Task 3 (Legal Information Retrieval) competitors of legal competition COLIEE in two years 2020 and 2021.
- Runner up prize in Legal Text Retrieval task Zalo AI competition in 2021.
- Organizing committee of the legal Workshop of KSE 2021: Automated Legal Question Answering Competition (ALQAC 2021).