

INDUCTIVE BIASES FOR DEEP LINGUISTIC STRUCTURED PREDICTION

by
Jie Cao

A dissertation submitted to the faculty of
The University of Utah
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Computer Science

School of Computing
The University of Utah
Dec 2021

Copyright © Jie Cao 2021

All Rights Reserved

The University of Utah Graduate School

STATEMENT OF DISSERTATION APPROVAL

The dissertation of Jie Cao
has been approved by the following supervisory committee members:

<u>Vivek Srikumar</u> ,	Chair(s)	<u>xx 2021</u> Date Approved
<u>Ellen Riloff</u> ,	Member	<u>xx 2021</u> Date Approved
<u>Qingyao Ai</u> ,	Member	<u>xx 2021</u> Date Approved
<u>Zac Imel</u> ,	Member	<u>xx 2021</u> Date Approved
<u>Yi Zhang</u> ,	Member	<u>xx 2021</u> Date Approved

by xx , Chair/Dean of
the Department/College/School of Computing
and by xx , Dean of The Graduate School.

ABSTRACT

Discovering the underlying structures from unstructured text will make sense of the rapidly growing data nowadays. Natural language, when described as a system of symbolic human communication, it is inherently studied via various symbolic and structured representations. We ground the studies of natural language structured prediction on both broad-coverage linguistic representations (such as graph-based meaning representations, DM, PSD, AMR, and UCCA) and application-specific representations (such as MISC code for psychotherapy dialog and dialog state representations for task-oriented dialog).

Towards generalization, the search for appropriate inductive biases is necessary for any machine learning (ML) based natural language processing (NLP) system, the same for predicting complicated combinatory structures from natural language via deep learning models. In this thesis, I primarily focused on two kinds of inductive biases on deep learning based linguistic structured prediction: *Structural Inductive Bias* and *Natural Language as Inductive Biases*.

Due to the compositionality of natural language, the underlying language representations are designed as compositional structures. I studied structural inductive biases via designing factorization-oriented learning and reasoning mechanisms on lexical, phrasal, and sentential levels. On the other side, much knowledge or human biases are encoded in the massive amount of human language. Taking natural language as a media of inductive biases, I studied how to leverage natural language to describe the functions of the intent/slot labels in task-oriented dialogue, thus guiding the model transferring to new domains and APIs with overlapping functions and task structures. The experimental results show our proposed methods achieve competitive performance on each task even with simple independent factorization assumptions, and they potentially help for other linguistic structured prediction tasks with similar inductive biases.

For my parents, and my love.

CONTENTS

ABSTRACT	iii
LIST OF FIGURES	viii
LIST OF TABLES	ix
NOTATION AND SYMBOLS	xii
CHAPTERS	
1. INTRODUCTION	1
1.1 Motivation	6
1.1.1 Generalization: The Need for Inductive Bias	6
1.1.2 Hypothesis Class and Hypothesis: The Origins of Inductive Biases	8
1.1.3 Inductive Biases for Deep Linguistic Structured Prediction	9
1.2 Contribution	11
1.3 Thesis Outline	13
2. BACKGROUND	15
2.1 Symbolic Representations for Natural Language	15
2.1.1 Broad-coverage Semantic Representation	15
2.1.1.1 Bi-lexical Semantic Dependency Parsing	16
2.1.1.2 Abstract Meaning Representation	16
2.1.1.3 Universal Conceptual Cognitive Annotation	16
2.1.2 Application-specific Representation: Dialog	16
2.1.2.1 Dialog Act	16
2.1.2.2 Dialog State Tracking	16
2.1.2.3 Conversational Semantic Representation	16
2.2 Deep Structured Prediction in NLP	16
2.2.1 Formulation of Structural Interdependence	17
2.2.1.1 Graphic Model	17
2.2.1.2 Declarative Interdependence	17
2.2.1.3 Learning Interdependence	17
2.2.2 Neural Representation Learning	17
2.2.3 Modeling Discrete Latent Variables	17
2.2.4 Other Advances in Structured Prediction	17
2.2.4.1 MAP Inference	17
2.2.4.2 Marginal Inference	17
2.3 Chapter Summary	18
3. PARSING MEANING REREPRESENTATIONS VIA LEXICAL AND PHRASAL ANCHORING	19

3.1	Related Work and Anchoring Analysis	20
3.1.1	Lexical-Anchoring Analysis on DM, PSD and AMR	20
3.1.1.1	Explicit Alignments: DM, PSD	21
3.1.1.2	Implicit Anchoring in AMR	21
3.1.1.3	Lexical-Anchoring	22
3.1.2	Analysis on Phrasal Anchoring in UCCA and TOP	22
3.1.3	Summary of Anchoring Analysis	23
3.2	Latent Alignment Model for Lexical-Anchoring Graph-based Parsing	24
3.2.1	Two-stage Graph-Based Parsing	24
3.2.1.1	Node Identification	25
3.2.1.2	Edge Identification	26
3.2.1.3	Inference	26
3.2.2	Latent Alignment Model	26
3.2.2.1	Continuous Relaxation for Discrete Alignments	27
3.2.2.2	VAE, Perturb-and-Map, Gumble Sinkhorn	27
3.3	Minimal Span-based CKY Parsing Framework	27
3.3.1	Graph to Constituent Tree Transformation	28
3.3.2	A Unified Span-based Model for CKY Parsing	29
3.3.2.0.1	Tree Factorization	29
3.3.2.0.2	CKY Parsing	30
3.3.3	Span Encoding	31
3.4	Experiments and Results	31
3.4.1	Dataset and Evaluation	31
3.4.2	Summary of Implementation	31
3.4.2.1	Top	32
3.4.2.2	Node	32
3.4.2.3	Edge	33
3.4.2.4	Connectivity	33
3.4.3	Model Setup	33
3.4.4	Results	34
3.4.4.0.1	Official Results on Lexical Anchoring	34
3.4.4.0.2	Official Results on Phrasal Anchoring	34
3.4.5	Error Breakdown	35
3.4.5.1	Error Analysis on Lexical-Anchoring	35
3.4.5.2	Error Analysis on Phrasal-Anchoring	36
3.5	Chapter Summary	37
4.	MODELING ON SENTENCE-ANCHORING FOR DIALOG IN THERAPY . . .	38
4.1	Background and Motivation	40
4.2	Task Definitions	41
4.3	Models for MISC Prediction	42
4.3.1	Encoding Dialogue	43
4.3.2	Word-level Attention	43
4.3.3	Utterance-level Attention	45
4.3.4	Predicting and Training	46
4.3.5	Addressing Label Imbalance	46
4.4	Experiments	47
4.4.1	Preprocessing and Model Setup	47

4.4.2	Results	48
4.4.3	Best Models	48
4.5	Analysis and Ablations	51
4.5.1	Label Confusion and Error Breakdown	51
4.5.2	How Context and Attention Help?	54
4.5.3	Can We Suggest Empathetic Responses?	55
4.6	Conclusion	55
5.	REPRESENTING INTENT/SLOT CONCEPT WITH NATURAL LANGUAGE DESCRIPTION	56
5.1	Schema-Guided Dialog State Tracking	58
5.2	Related Work	59
5.3	Datasets	60
5.4	Dialog & Schema Representation and Inference (Q1)	62
5.4.1	Encoder Architectures	62
5.4.2	Model Overview	64
5.4.3	Experiments on Encoder Comparison	65
5.5	Supplementary Training (Q2)	67
5.5.1	Intermediate Tasks	68
5.5.2	Results on Supplementary Training	68
5.6	Impact of Description Styles (Q3)	69
5.6.1	Benchmarking Styles	69
5.6.2	Results on Description Styles	70
5.6.2.1	Homogeneous Evaluation	70
5.6.2.2	Heterogeneous	72
5.7	Conclusion	73
6.	CONCLUSIONS AND FUTURE WORK	75
6.1	Claims and Research Contribution Revisited	75
6.2	Future Work Direction	75
6.3	Summary	75
APPENDICES		
A.	THE FIRST	76
B.	THE SECOND	77
C.	THE THIRD	78
REFERENCES		79
BINOMIAL NOMENCLATURE INDEX		91
TOPIC INDEX		92

LIST OF FIGURES

1.1	The part-of-speech for the sentence " <i>The dog cannot find the bone it hid from the other dogs</i> "	1
1.2	The constituent tree [<i>redraw trees_{TODO}</i>] for the sentence " <i>The dog cannot find the bone it hid from the other dogs</i> "	2
1.3	The dependency tree, for the sentence " <i>The dog cannot find the bone it hid from the other dogs</i> "	2
1.4	[<i>SDP_{TODO}</i>] The broad coverage meaning representation AMR for the sentence " <i>The dog cannot find the bone it hid from the other dogs</i> "	4
1.5	[<i>SDP_{TODO}</i>] The broad coverage meaning representation UCCA for the sentence " <i>The dog cannot find the bone it hid from the other dogs</i> "	4
1.6	[<i>dialog act and dialog state tracking example_{TODO}</i>]	5
1.7	Hypothesis, Generatioalization in Supervised Learning Setting[<i>Current fig is directly copied from Ben Taskar's thesis, redraw it_{TODO}</i>]	7
3.1	Architecture of graph-based model and inference, for running exmaple [wsj#0209013]	25
3.2	UCCA to Constituent Tree Transformation for [wsj#0209013]	29
3.3	TOP to Constituent Tree Transformation for the utterance " <i>Driving directions to the Eagles game</i> "	30
4.1	Confusion matrix for categorizing client codes, normalized by row.	52
4.2	Confusion matrix for categorizing therapist codes, normalized by row.	53
5.1	An example dialog from Restaurant_1 service, along with its service/intent/slot descriptions and dialog state representation.	57
5.2	Dual-Encoder, Cross-Encoder and Fusion Encoder, shaded block will be cached during training	63

LIST OF TABLES

3.1	Detailed classifiers in our model, round bracket means the number of output classes of our classifier, * means copy mechanism is used in our classifier. At the end of shared task, EDS are not fully supported to get an official results, we leave it as our future work.	31
3.2	Detailed classifiers in our model, round bracket means the number of output classes of our classifier, * means copy mechanism is used in our classifier. At the end of shared task, EDS are not fully supported to get an official results, we leave it as our future work.	32
3.3	Official results overview on unified MRP metric, we selected the performance from top 1/3/5 system(s) for comparison	34
3.4	Official results overview on unified MRP metric, we selected the performance from top 1/3/5 system(s) for comparison. It shows our UCCA model for post-evaluation can rank 5th	35
3.5	Our parser on AMR ranked 1st. This table shows the error breakdown when comparing to the baseline TUPA model and top 2 Che et al. (2019) in official results	35
3.6	Our parser on DM ranked 7th. This table shows the error breakdown when comparing to the model ranked Top 1 Li et al. (2019) in official results	36
3.7	Our parser on PSD ranked 6th. This table shows the error breakdown when comparing to the model ranked top 1 Donatelli et al. (2019) in official results	36
3.8	Our UCCA parser in post-evaluation ranked 5th according to the original official evaluation results. This table shows the error breakdown when comparing to the model ranked top 1 Che et al. (2019) in official results. * denotes the ranking of post-evaluation results	37
4.1	Distribution, description and examples of MISC labels.	39
4.2	An example of ongoing therapy session	41
4.3	Summary of word attention mechanisms. We simplify BiDAF with multiplicative attention between word pairs for f_m , while GMGRU uses additive attention influenced by the GRU hidden state. The vector $w_e \in \mathbb{R}^d$, and matrices $W^k \in \mathbb{R}^{d \times d}$ and $W^q \in \mathbb{R}^{2d \times 2d}$ are parameters of the BiGRU. The vector h_{j-1} is the hidden state from the BiGRU in GMGRU at previous position $j - 1$. For combination function, BiDAF concatenates bidirectional attention information from both the key-aware query vector a_{ij} and a similarly defined query-aware key vector a' . GMGRU uses simple concatenation for f_c	45
4.4	Input options for annotating and forecasting tasks based on CON and HGRU skeletons.	46

4.5	Main results on categorizing client codes, in terms of macro F_1 , and F_1 for each client code. Our model \mathcal{C}_C uses final dialogue vector H_n and current utterance vector v_n as input of MLP for final prediction. We found that predicting using $\text{MLP}(H_n) + \text{MLP}(v_n)$ performs better than just $\text{MLP}(H_n)$	49
4.6	Main results on categorizing therapist codes, in terms of macro F_1 , and F_1 for each therapist code. Models are the same as Table 4.5, but tuned for therapist codes. For the two grouped MISC set MIA and MIN , their results are not reported in the original work due to different setting.	49
4.7	Main results on forecasting client codes, in terms of F_1 for ST , CT on dev set, and macro F_1 , and F_1 for each client code on the test set.	50
4.8	Main results on forecasting therapist codes, in terms of Recall@3, macro F_1 , and F_1 for each label on test set	50
4.9	Categorization of CT / ST confusions. The two numbers in the brackets are the count of errors for predicting CT as ST and vice versa. We exemplified 100 examples for each case.	52
4.10	Ablation study on categorizing client code. * is our best model \mathcal{C}_C . All ablation is based on it. The symbol + means adding a component to it. The default window size is 8 for our ablation models in the word attention and sentence attention parts.	54
4.11	Ablation study on categorizing therapist codes, * is our proposed model \mathcal{C}_T . \ means substituting and – means removing that component. Here, we only report the important REC , RES labels for guiding, and the MIN label for warning a therapist.	55
5.1	Summary of characteristics of SG-DST MULTIWOZ 2.2 datasets, in domain diversity, function overlap, data collecting methods	60
5.2	Schema description input used for different tasks to compare <i>Dual-Encoder</i> , <i>Cross-Encoder</i> , and <i>Fusion-Encoder</i> . In the appendix ??, we also studies other compositions of description input. We found that service description will not help for INTENT, REQ and CAT tasks, while the impact on NONCAT task also varies from SG-DST and MULTIWOZ 2.2 dataset.	66
5.3	Test set results on SG-DST and MULTIWOZ 2.2. The <i>Dual-Encoder</i> model is a re-implementation of official DSTC8 baseline from Rastogi et al. (2019). Other models are trained with the architecture described in our paper.	66
5.4	Relative performance improvement of different supplementary training on SG-DST dataset	67
5.5	Relative performance improvement of different supplementary training on MULTIWOZ 2.2 dataset.	67
5.6	Homogeneous evaluation results of different description style on SG-DST dataset and MULTIWOZ 2.2 datasets. The middle horizontal line separate the two name-based descriptions and two rich descriptions in our settings. All numbers in the table are mixed performance including both seen and unseen services.	71

5.7	Performance changes when using BERT finetuned on SQuAD2 dataset to further finetuning on our NONCAT task.	72
5.8	Results on unseen service with heterogeneous description styles on SG-DST dataset. More results and qualitative analysis are in the appendix ??.....	73

NOTATION AND SYMBOLS

α	fine-structure (dimensionless) constant, approximately 1/137
α	radiation of doubly-ionized helium ions, He ⁺⁺
β	radiation of electrons
γ	radiation of very high frequency, beyond that of X rays
γ	Euler's constant, approximately 0.577 215 ...
δ	stepsize in numerical integration
$\delta(x)$	Dirac's famous function
ϵ	a tiny number, usually in the context of a limit to zero
$\zeta(x)$	the famous Riemann zeta function
...	...
$\psi(x)$	logarithmic derivative of the gamma function
ω	frequency

CHAPTER 1

INTRODUCTION

During the daily interactions of our social, entertainment, and professional life, human language is essential for human intelligence, and it is arguably our most powerful tool for the learning and transmission of knowledge. With the advances of computers and the internet, most of the world's knowledge, such as conversations, scholarly research, factual news, online education, and private mental health records, is now easily accessible as digitized text. However, with very limited ability of information processing, human cannot easily find the hidden knowledge from the huge amount of unstructured text.

One classical way to study unstructured natural language is to represent the language via various structured symbolic representations in different levels. Before the revolution of representation learning with deep learning, the NLP community has put decades of effort into solving different linguistic structured prediction tasks in a pipeline, to get various aspects of text understanding. For examples:

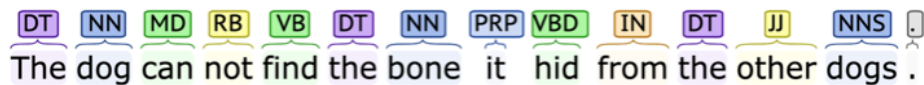


Figure 1.1: The part-of-speech for the sentence "The dog cannot find the bone it hid from the other dogs"

Example 1: Part-of-Speech Tagging, Consistency and Dependency Structure. As shown in Figure 1.1, the part-of-speech (POS) tagging assigns each word in a sentence a part-of-speech tag, such NOUN, VERB, ADJECTIVE, PRONOUN, etc. How to capture the sequential structure of correlations between consecutive tags is the key of this task. Figure 1.2 shows the **constituent tree** structure of the sentence. The constituent tree parsing is to recognize the recursive phrase structure of a sentence, such as noun, verb, prepositional phrases, and their nesting in relation to each other. Figure 1.3 shows the **dependency tree** structure of the sentence. Instead of the constituency structure, here the syntactic structure of a sen-

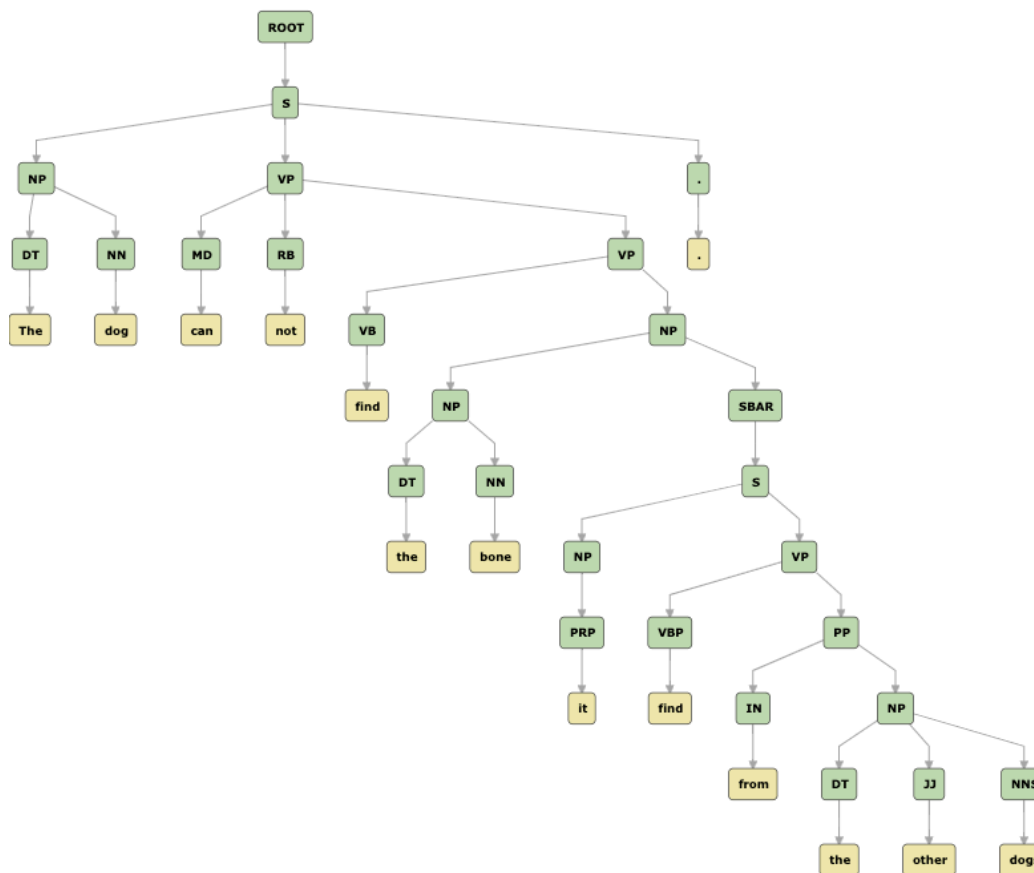


Figure 1.2: The constituent tree [redraw trees_{TODO}] for the sentence "The dog cannot find the bone it hid from the other dogs"

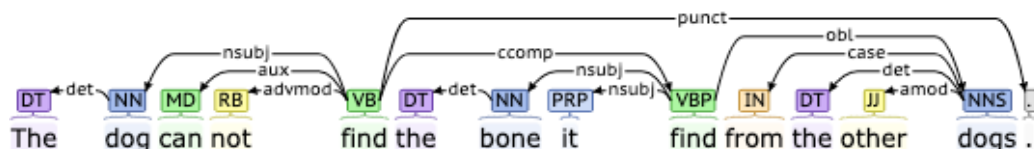


Figure 1.3: The dependency tree, for the sentence "The dog cannot find the bone it hid from the other dogs"

tence is described in terms of the directed bi-lexical grammatical relations between words. Each labeled arcs represents a directed relation from heads to dependents. Besides the above lexical and syntactic structured information, various semantics in natural language are also widely studied in structured representation, such as **named entity recognition**, **coreference resolution**, **semantic role labelling**¹. Especially, the above structured information are widely used as features in machine learning based NLP system.

¹More details about various semantic phenomenos will be introduced in Section 2.1.1

Example 2: Broad Coverage Meaning Representation. Besides the above structures capturing specific lexical, syntactic or semantic information, recent developments in semantic representation aims to represent the multiple phenomenos in a single structure. As shown in Figure 1.4, it shows the Abstract Meaning Representation (**AMR**, Banarescu et al., 2013a) of the corresponding sentence. The node in the graph represent the abstract concept², and the labeled edges between the nodes represent the relations between those concepts. Figure 1.5 shows the foundational layer of Universal Conceptual Cognitive Annotation (**UCCA**, Abend and Rappoport, 2013a), which is a multi-layered framework for semantic representation that aims to accommodate the semantic distinctions expressed through linguistic utterances, and support open-ended extension. This foundational layer focuses on argument structures of verbal, nominal and adjectival predicates and the inter-relations between them.

Example 3: Application-specific Symbolic Representation. Besides the above broad coverage syntactic and semantic structures in natural language, researchers have designed various symbolic representation for specific applications. Take dialogue application as an example, dialog act and dialog intents are firstly desgined to represent the speech act or intention of each utterance. To represent the state of the dialog, inspired by the case theory, frame-based representation in GUS (Bobrow et al., 1977) are introduced into dialog state representation, which consists of a collection of slots and each with a set of possible values. Figure 1.6 shows an example of dialog state tracking, where each table filling with intent/slot/slot values represent a dialog state for a user turn.

From specific linguistic structures to broad coverage semantic representation and application-specific representations, they are interpretable to both human and computers. Exploit this structured analysis on symbolic representation can offer rigorous document analysis, easier knowledge organization, programmable reasoning, which are also potentially helpful to offer actionable suggestions to guide human behavior, such as mental health (Tanana et al., 2016), dialog state tracking (Budzianowski et al., 2018), scientific document analysis (Dernoncourt and Lee, 2017), and so on.

With the stunning rise of deep learning, modern NLP systems have achieved outstand-

²AMR concepts includes PropBank framesets, and other sepcial date, and spatial entities, etc. More details about AMR will be introduced in Section 2.1.1

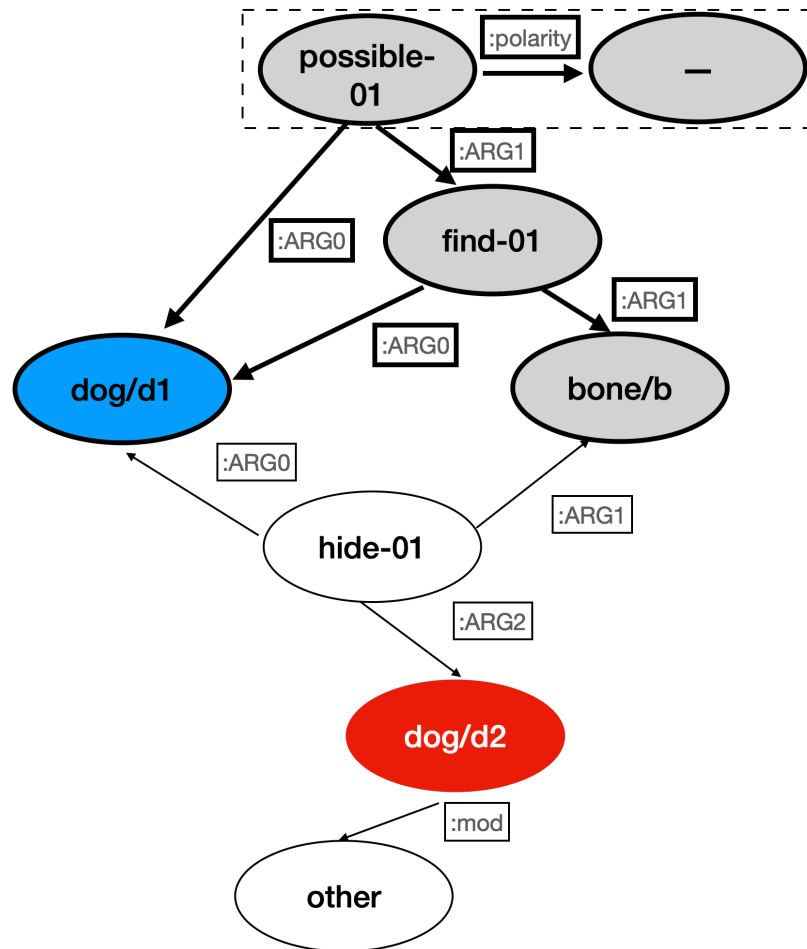


Figure 1.4: [SDP_{TODO}] The broad coverage meaning representation AMR for the sentence "The dog cannot find the bone it hid from the other dogs"

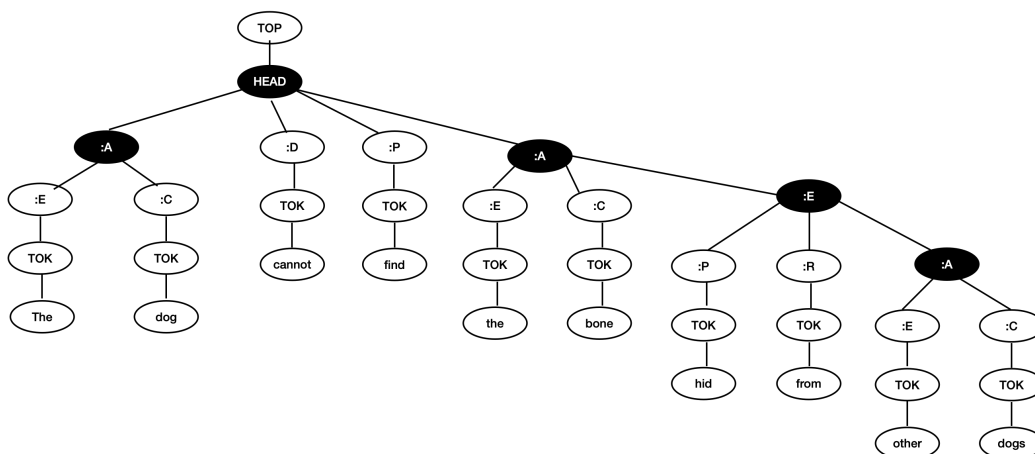
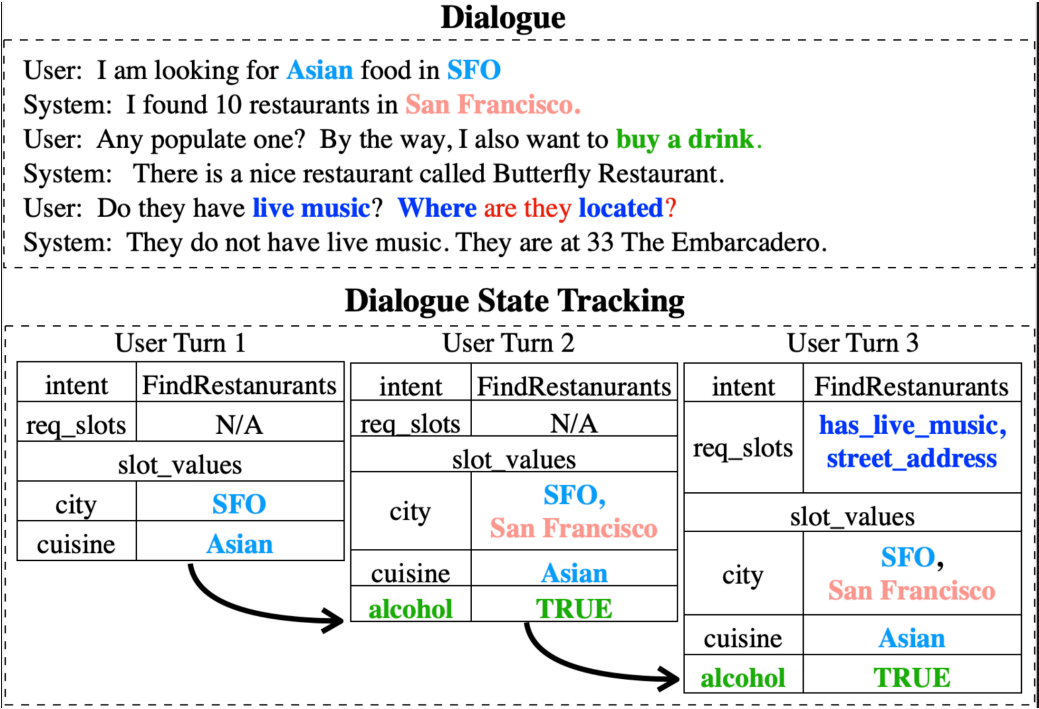


Figure 1.5: [SDP_{TODO}] The broad coverage meaning representation UCCA for the sentence "The dog cannot find the bone it hid from the other dogs"



Hence, from classical NLP pipeline to modern deep learning based models, predicting the underlying structured representations are arguably important for natural language processing, and still important in deep learning era. In this thesis, we ground the studies of natural language structured prediction on both broad-coverage linguistic representations (such as graph-based meaning representations, DM, PSD, AMR, and UCCA, etc.) and application-specific representations (such as MISC code for psychotherapy dialog and dialog state representations for task-oriented dialog). More importantly, considering both the success and the failure for the pure data-driven deep learning model, we focus our study on the inductive biases useful in deep linguistic structured prediction tasks.

1.1 Motivation

In this section, we first study the need of inductive biases in machine learning. Then we analyze where the current deep learning models can get inductive biases from, and finally we highlight some open problems that this thesis addresses about inductive biases on deep linguistic structured prediction.

1.1.1 Generalization: The Need for Inductive Bias

Any system (natural or artificial) that makes general inferences on the basis of particular and limited data must constrain its hypotheses in some way before observing data. With limited observations and resources (time, memory, energy), our human intelligence of generalizing to new environments makes us efficiently learn when interacting with the world and other human beings. This efficiency largely depends on many **inductive biases** from human intelligence (Gershman, 2021), which are potentially helpful for machine intelligence. According to extensive cognitive studies (Spelke, 1990; Bienenstock et al., 1996; Rehder, 2003; Harlow, 1949; Lake et al., 2016; Gershman, 2021), there are many key inductive biases for human intelligence, such as compositionality, causality, learning to learn, and so on. We are not meaning machine intelligence should mimic the human intelligence. Instead, we arguably that mean those key human inductive biases under limited observation and resources may inspire us to design the machine intelligence.

On the machine intelligence side, as the no-free-lunch theorem for machine learning said (Baxter, 2000; Wolpert et al., 1995), inductive biases that influence hypothesis selec-

tion is necessary to obtain generalization. Tom M Mitchell argues that inductive biases constitute the heart of generalization and indeed a key basis for learning itself (Mitchell, 1980).

As the supervised learning setting shown in Figure 1.7, \mathcal{H} refers to machine learning model families, including those deep learning models. According to the universal approximation theorem, properly parameterized neural network can represent any functions. The task of finding a hypothesis h is reduced to estimating the neural network parameters by fitting the training data. However, the labeled data is usually noisy and cannot represent the whole distribution of the real data distribution, let alone new input data with distribution shift. Hence, preferences on how to choose the hypothesis class \mathcal{H} and how to select the h is necessary to generalize to new data.

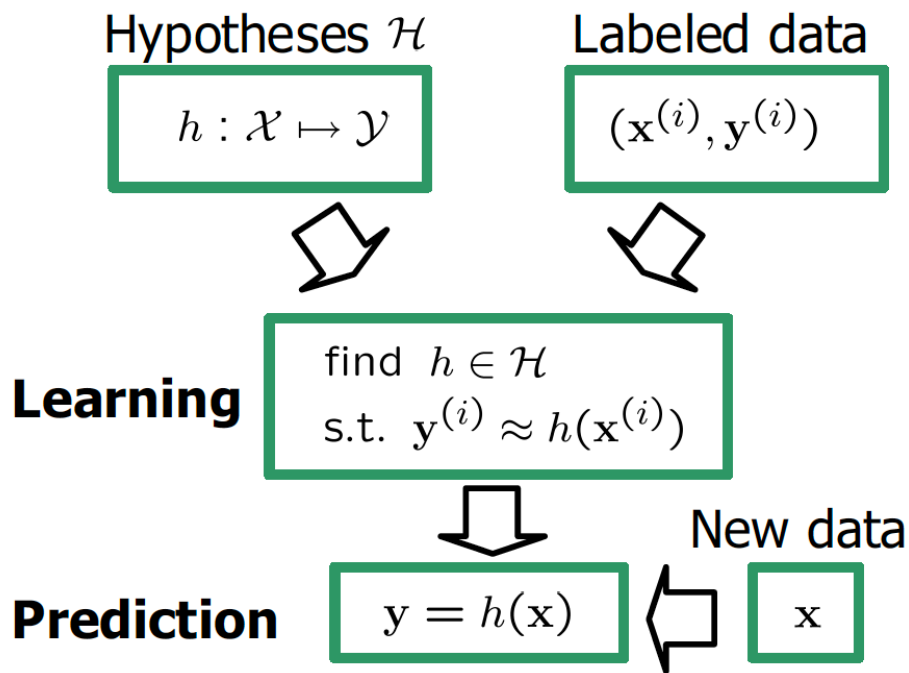


Figure 1.7: Hypothesis, Generalization in Supervised Learning Setting[Current fig is directly copied from Ben Taskar's thesis, redraw it_{TODO}]

In this thesis, following the definition of *bias* in Mitchell (1980), we refer *inductive bias* as follows:

Definition of Inductive Bias. 'Any biases for choosing one generalization over another, other than strict consistency with the observed training instances'. □

1.1.2 Hypothesis Class and Hypothesis: The Origins of Inductive Biases

Inductive biases are widely used in the whole history of machine learning. As the supervised learning setting in Figure 1.7 and the definition inductive biases in Figure 1.1.1, inductive biases are mainly from hypothesis class selection and how to find the specific hypothesis h .

According to the two goals, we organize the origins of inductive biases in two parts:

Goal 1: Deciding hypothesis class \mathcal{H} .

- **Model Family** Different model family can represent different hypothesis classes. For example, generalized linear models such as logistic regression and support vector machines, can only support arbitrary linear decision boundaries. [add more models`TODO`].
- **Feature Engineering and Kernel Methods** [Inductive biases also can be found in feature engineering, `TODO`] However, not every training data are linear separable. One solution to this kernel-based SVM. The design of kernel is inductive bias.
- **Neural Architecture and Representation Learning** What's more, inductive bias are widely used in modern deep learning architectures, which also decides the hypothesis space. For example, translation invariance for convolutional neural networks (CNN) and pooling, and recurrent assumption of recurrent neural networks (RNN), equivariance over permutation for neural graph networks (GNN), positional encoding for word orders in Transformer. [geometry deep learning`TODO`]

Goal 2: Finding the specific hypothesis h .

- **Optimization** Even in existed computing-heavy and data-heavy scenarios, inductive biases are widely used in well-known deep learning techniques. For example, smoothness assumption in optimization method, such as Stochastic gradient decent, which was proofed as another inductive bias to have better generalization. [SGD generalization, smoothness`TODO`] [Optimization will control the bias shift during learning`TODO`]
- **Training Data** It is often the case that available datasets do not exactly represent the data distribution of interest. One particularly problematic case is when the dataset

is biased in some way against a particular demographic group, which often leads to model predictions that unfairly disadvantage members of that group.

[Data augmentation for different perturbation_{TODO}]

[Realistic distribution shift_{TODO}]

- **Inference Algorithm** Combinatorial Optimization approaches, such as graph cuts, partitions, bipartite matching, dynamic programming can also be involved during the hypothesis learning, which will also constrain the learning. In this thesis, we mainly focus on representation learning, for inference, we use the existing methods, such as greedy, maximum spanning connected graph, dynamic programming for CKY parsing.

In this thesis, we mainly study inductive biases from neural architecture and the corresponding representation learning.

1.1.3 Inductive Biases for Deep Linguistic Structured Prediction

Towards flexible out-of-distribution and systematic generalization, the search for appropriate inductive biases is also necessary for deep linguistic structured prediction.

Given an observation $x \in \mathcal{X}$ (natural language text), define a structured prediction $y \in \mathcal{Y}(x)$ as $y = f(x)$. Here y is a structured symbolic representation for x , e.g., y could be a sequence of part-of-speech tagging, constituent tree or dependency tree, or y can be a broad-coverage meaning representation, like AMR, UCCA, or an application-specific representation like frame-based dialog state Bobrow et al. (1977). To represent the target function $f(x)$, we adopt the popular energy-minimization strategy by defining $f(x)$ as the minimizer of an auxiliary energy optimization problem.

$$f(x) = \arg \min_{y \in \mathcal{Y}(x)} E(x, y), \quad (1.1)$$

where $E(x, y)$ is a scoring function to represent the energy between x and different candidate output structures y .

In many NLP applications, the candidate output set $\mathcal{Y}(x)$ is typically finite but exponentially large, and its size may depend on the input x . To render either exact or approximate optimization of energy-minimization in Equation 1.1, how to model the representation of x and y , and the interactions between them are challenging problems. Practitioners

typically employ energy functions with specific factorization structures to design efficient algorithms, by assuming the whole energy $E(x, y)$ can be decomposed as a sum of **factors** $c \in C$. A popular choice to represent the facortization is to index both x and y as a set of sub components $x = (x_1, \dots, x_i, \dots, x_N)$ and $y = (y_1, \dots, y_j, \dots, y_M)$.

In AMR parsing as shown in Figure 1.4, x_i can be each word or multi-word expression in a sentence, while y_j can be single AMR node and relation. For dialog state tracking in Figure ??, x_i is an utterance in the dialog, while y_i is the value for each intent and slot in the predicted frames.

The interdependence assumptions between those subcomponents in x and y are key in structured prediction model. In the chapter 2, we show various representation formalism (such as graphic models), structured learning (max-margin framework) and inference approaches (dynamic programming, integer linear programming) to modeling the interdependence. Inductive biases can be designed in all the three key aspects.

In this theis, for representation formalism for factorization assumption, we always assume the independent factorization paradigm, where each factor c only depends on a subset of y components y_c and the aligned x components (anchors) $x_c = a(y_c)$. In another word, the output part y_c in factor c is independent with output part $y_{c'}$ in another factor c' . Here, $a(y_c)$ is the alignment model to find how independent output parts y_c are anchored to the constituents of the observation x , thus the prediction of each y_c are independent with each other, and can be locally decided by its aligned anchors.

$$\begin{aligned}
 E(x, y) &= \sum_{c \in C} E(x, y_c) = \sum_{c \in C} E(a(y_c), y_c) \\
 f(x) &= \arg \min_{y \in \mathcal{Y}(x)} E(x, y), \\
 &= \arg \min_{y \in \mathcal{Y}(x), a \in A} E(a, x, y),
 \end{aligned} \tag{1.2}$$

Hece, this simple independent factorization decompose the structured learning into decomposed local learning. More importantly, it also makes the inference intractable, thus can be easily involved in the end-to-end neural network training.

Using the AMR Paring in ?? as an continue example, by allowing the input constituent x_c mapped into an empty output $y_c = \phi$, the design of independent factorization will first segment the output y into small parts $y_c \in \text{seg}_{out}(y)$, then find the anchors x_c for each

y_c from the candidate decomposition set $seg_{in}(x)$. For example, one of the segmented y_c in Figure ?? is a pre-categorized sub-graph ‘(possible-01 :polarity -)’, and $a(y_c)$ of that is the anchor word ‘cannot’. However, the words ‘the’, ‘from’ are mapped to empty nodes. During inference, we can directly produce y_c for all candidates segments in a new sentence as shown in Figure ?. We first prepatate a list of candidate anchors $seg_{in}(x) = \{\text{‘The’}, \text{‘dog’}, \text{‘found’}, \text{‘the’}, \text{‘bone’}, \text{‘it’}, \text{‘hide’}\}$, then the model will produce the independent prediction y_c of each anchor as $\{\phi, \text{‘dog’}, \text{‘find-01’}, \phi, \text{‘bone’}, \text{‘it’}, \text{‘hide’}\}$, then we assemble the non-empty y_c by predicing the relations between each other and finally forms y via postprocessing³.

Hence, in the independence factorization setting, the problem is reduced into three challenges:

- How to decompose the output y into a set of independent parts y_c .
- How to derive the aligned input a_{y_c} .
- How to model the representation of a_{y_c} and y_c , thus computing the energy score $E(x_{a(y_c)}, y_c)$.

In this thesis, the first question on independently decomposing y is either straightforward or has been resolved by previously existing methods in our studied tasks. We mainly focus on the remaining challenges on modeling alignment and representation learning.

1.2 Contribution

Thesis Statement. Our claim is that by designing *Structural Inductive Bias* and *Natural Language as Inductive Biases*, models with naive independent factorization can achieve strong performance on predicting the structures of natural language.

Contribution. In this theis, focusing on independent factorization setting, we show that our proposed indutive biases can offer discriminative features to improve the performance of the locally dependent output structured prediction model, and still achieve competetive and generalizable performance in two sets of tasks, including broad-coverage meaning representation and application-specific representations.

³The post-processing include mergeing coreference nodes (as the ‘dog’ and ‘it’), adding other attributes

We next summarize the main contributions of this thesis, addressing some of the open problems mentioned in the previous section.

1. We proposed a unified parsing framework to support both **explicit lexical-anchoring** (DELPH-IN MRS Bi-lexical Dependencies (DM, Ivanova et al., 2012) and Prague Semantic Dependencies (PSD, Hajic et al., 2012; Miyao et al., 2014)), and **implicit lexical anchoring** (AMR). Over 16 teams in the shared tasks, my parser Cao et al. (2019b) *ranked 1st on AMR, 6th in DM, 7th in PSD*. By combining Perturb-and-MAP sampling with differentiable Gumbel-Softmax Sinkhorn Networks, we can approximately infer the discrete latent-alignment variable in lexical-anchoring of the independent factorization setting.
2. Notice that the **phrasal-anchoring** Universal Conceptual Cognitive Annotation (UCCA, Abend and Rappoport, 2013b) and Task-oriented Dialog Parsing (TOP, Gupta et al., 2018) is similar to constituency tree structure, except for unseen phenomena such as remote edges and discontinuous spans, we extrapolate the existed algorithmic inductive bias on tree structure prediction and Cost-augmented CKY inference to the new UCCA and TOP parsing tasks. Powered by strong span-representation learning method, my system (Cao et al., 2019b) *ranked 5/16 on UCCA parsing*, and it can be reused to TOP parsing after a few preprocessing steps, and outperform several baseline models.
3. We address the problem of providing real-time guidance to therapists with a dialogue observer. It decomposes the dialog structure analysis with two independent factorization tasks: (1) categorizes therapist and client MI behavioral codes and, (2) forecasts codes for upcoming utterances to help guide the conversation and potentially alert the therapist. For both tasks, I studied a hierarchical gated recurrent unit (HGRU) with the *word-level attention* and *sentence-level attention* to distinguish different importance of words and sentences to our prediction Cao et al. (2019a). Our experiments demonstrate that our models can outperform several baselines for both tasks. We also report the results of a careful analysis that reveals the impact of the various network design tradeoffs for modeling therapy dialogue.

4. Natural language can be leveraged as inductive biases to describe the functions of the intent/slot labels in task-oriented dialogue. We are among the first to use large pretrained language models on description-based dialog state tracking, we offer detailed comparative studies how to transfer inductive biases to new domains and APIs with overlapping functions and task structures, including encoding strategies, supplementary pretraining, homogenous and heterogeneous evaluations.

1.3 Thesis Outline

To coherently present the thesis, we think it is helpful to discuss prior work related to an application in its own chapter, instead of putting them all in a single chapter. Therefore, we include a section of related work at the end of each application chapter. This thesis is divided into five parts, which we described below.

Chapter 2. Background. The first part systematizes the background of this thesis study in two sections:

- *Structures in NLP.* We first provide the necessary background about structures in natural languages for better exposing our contributions in remaining chapters.
- *Structured Prediction, Learning and Inference.* We summarize the recent advances in deep structured prediction with respect to learning and inference respectively. Especially, we overview the development of representation learning methods for natural language, from feature selection to deep learning based representation learning methods.

Chapter 3. Structural Inductive Biases for Lexical and Phrasal Anchoring. In this chapter, we provide lexical and phrasal anchoring analysis to decompose the output structures into locally dependent parts, where each part can be derived from its anchoring words or phrasal in the input sentence. For lexical-anchoring, we propose a unified model to support both explicit and implicit alignment information between each input parts and output parts. For phrasal-anchoring, we compared different ways to learn the contextualized representation to represent the span, and how they can bring discriminative features to our locally-dependent model. We show that with the above lexical and phrasal-anchoring based structural inductive biases for energy factorization and contextualized representa-

tion learning, our model can learn efficient discriminative features for the anchor representations and still achieve high performance in the locally-dependent model.

Chapter 4. Structural Inductive Biases for Sentence and Dialog. In the following two chapters, we extend our study to structures beyond a single sentence. In this chapter, We study the sequential dialog flow structure in a style of therapy called Motivational Interviewing (MI, Miller and Rollnick, 2003, 2012), which is widely used for treating addiction-related problems. Sentence-level tags called Motivational Interview Skill Codes are designed to represent the intention of each utterance and the dialogue flow of the whole therapy session. By developing a modular family of neural networks categorizing and forecasting the dialog flow in the form of MISC codes, we show that the above mechanisms on dialogue representation can efficiently model the sequential structure of dialogue flow.

Chapter 5. Natural Language as Inductive Biases. In this chapter, we study using natural language descriptions to represent the meaning of output symbols (Intents and Slots) in task-oriented dialog state tracking, which helps to reduce the poor scalability to transfer to unseen domain and services. We study three main challenges of using natural language for label representation: schema encoding, supplementary training, and description styles.

Chapter 6. Conclusion and Future Work. This chapter concludes, by providing a summary of contributions and drawing possible directions of future work.

CHAPTER 2

BACKGROUND

In the following two sections, we will briefly review the background on symbolic representation for natural language including broad-coverage meaning representations and application-specific representations. Then, we will give an overview of the linguistic structure prediction and recent advances on deep structured prediction.

2.1 Symbolic Representations for Natural Language

This is a chapter to introduce the background of meaning representations and related works on each of them.

2.1.1 Broad-coverage Semantic Representation

For linguistic analysis, structures have been studied from subword-level morphology Beesley and Karttunen (2003), word-level lexicon semantics Miller (1998), to single-sentence syntax/semantic representations Baker et al. (1998); Palmer et al. (2005); Collins (2003), and multi-sentences discourse analysis Carlson et al. (2003); Wolf and Gibson (2005); Prasad et al. (2008). This thesis covers broad-coverage graph-based semantic representations in different anchoring types. For lexical anchoring, we cover the DELPH-IN MRS Bi-lexical Dependencies (DM, Ivanova et al., 2012) and Prague Semantic Dependencies (PSD, Hajic et al., 2012; Miyao et al., 2014), Abstract Meaning Representation (AMR, Banarescu et al., 2013a); While for phrasal-anchoring, we study Universal Conceptual Cognitive Annotation (UCCA, Abend and Rappoport, 2013a).

2.1.1.1 Bi-lexical Semantic Dependency Parsing

2.1.1.2 Abstract Meaning Representation

2.1.1.3 Universal Conceptual Cognitive Annotation

2.1.2 Application-specific Representation: Dialog

In this thesis, we ground the study on application-specific representation on dialogue. In utterance-level, dialogue acts are designed to represent the function of each utterance in the dialogue. Similarly, Motivational Interview Skill Codes (MISC, Miller and Rollnick, 2003, 2012) are also proposed to represent the functions of each client and therapist utterance in the psychotherapy dialogue. Frame-based representation for dialogue is popular in modern dialogue systems, where the state of the dialogue is represented as the combinations of intent and slots. Recently, conversational parsing has attract much attention to represent the dialogue state in a more compositional way, such as the hierarchical tree structure in the Task-Oriented dialogue Parsing (TOP, Gupta et al., 2018) and so on.

2.1.2.1 Dialog Act

2.1.2.2 Dialog State Tracking

2.1.2.3 Conversational Semantic Representation

2.2 Deep Structured Prediction in NLP

Due to the power of representation learning, deep learning is widely used to extract sophisticated representations for the inputs in various NLP tasks. In this proposal, instead of focusing on a single task, we systematically study the representation learning challenges for multiple sets of tasks based on different factorization strategies. Besides the representation learning for sequential input text, we also study the representation of output structures. It is related to the main research problem for efficient deep structured inference (for both training-time and test-time inference): designing continuous and differentiable relaxations for *argmax* inference. Most works studied how to inject known hand-crafted constraints between discrete variable assignments, e.g., using differentiable fuzzy logic operators Li and Srikumar (2019). By relaxing the whole structured outputs into continuous vectors, structured prediction energy networks (SPEN, Belanger and McCallum, 2016) define energy functions that can learn the arbitrary dependencies among parts of structured outputs. Following this, inference network Tu and Gimpel

(2018) was proposed to approximate the cost-augmented inference during training and then fine-tuning for test-time inference.

In this thesis, we study the specific inductive biases based on the principle of compositionality, which can be used for **different purposes more than the accuracy on a single task or a single domain**. For example, with simple independent factorization, we study a universal alignment-based model to support *cross-framework* meaning representation parsing. By defining two complementary dialogue observers to sequentially predict the MISC code for both current and future utterance, our model emphasizes both *accurate and real-time* assistance to a therapist. We propose to represent the output labels with natural language descriptions for *zero-shot learning* on unseen labels §??. In our thesis, we show that during the rapid progress of representation learning methods, our proposed inductive biases still can outperform the standard usage baselines.

2.2.1 Formulation of Structural Interdependence

2.2.1.1 Graphic Model

2.2.1.2 Declaritive Interdependence

2.2.1.3 Learning Interdependence

2.2.2 Neural Representation Learning

For deep learning based methods, we review the recent advances from embedding based methods to attention-based dynamic features selection. Finally, we also introduced the rapid progress in language encoding architectures, from recurrent neural network to transformer, and the corresponding pretrained language models ELMo , BERT, GPT3 etc.

2.2.3 Modeling Discrete Latent Variables

2.2.4 Other Advances in Structured Prediction

2.2.4.1 MAP Inference

Exact: Dynamic Programming: Viterbi, CKY, Max Spanning Arborescence, Exact Search, Integer Linear Programming Approximate: Sampling, Beam Search/Learning to Search, Linear Programming Relaxations

2.2.4.2 Marginal Inference

Forward-Backward/ Inside-outside/ Matrix tree CRF SparseMAP

2.3 Chapter Summary

Instead of studying the structured prediction for each of them separately, we categorize them according to different factorization patterns, and then we study the factorization-oriented representation learning methods for each of them.

CHAPTER 3

PARSING MEANING REREPRESENTATIONS VIA LEXICAL AND PHRASAL ANCHORING

The design and implementation of broad-coverage and linguistically motivated meaning representation frameworks for natural language is attracting growing attention in recent years. With the advent of deep neural network-based machine learning techniques, we have made significant progress to automatically parse sentences into structured meaning representation (Oepen et al., 2014, 2015; May, 2016; Hershcovich et al., 2019). Moreover, the differences between various representation frameworks has a significant impact on the design and performance of the parsing systems.

Due to the abstract nature of semantics, there is a diverse set of meaning representation frameworks in the literature (Abend and Rappoport, 2017). In some application scenario, tasks-specific formal representations such as database queries and arithmetic formula have also been proposed. However, primarily the study in computational semantics focuses on frameworks that are theoretically grounded on formal semantic theories, and sometimes also with assumptions on underlying syntactic structures.

Anchoring is crucial in graph-based meaning representation parsing. Training a statistical parser typically starts with a conjectured alignment between tokens/spans and the semantic graph nodes to help to factorize the supervision of graph structure into nodes and edges. In this chapter, with evidence from previous research on AMR alignments (Pourdamghani et al., 2014; Flanigan et al., 2014; Wang and Xue, 2017; Chen and Palmer, 2017; Szubert et al., 2018; Lyu and Titov, 2018), we propose a uniform handling of three meaning representations from Flavor-0 (DM, PSD) and Flavor-2 (AMR) into a new group referred to as the **lexical-anchoring** MRs. It supports both explicit and implicit anchoring of semantic concepts to tokens. The other two meaning representations from Flavor-1 (EDS, UCCA) is referred to the group of **phrasal-anchoring** MRs where the

semantic concepts are anchored to phrases as well.

To support the simplified taxonomy, we named our parser as LAPA (**L**exical-**A**nchoring and **P**hrasal-**A**nchoring)¹. We proposed a graph-based parsing framework with a latent-alignment mechanism to support both explicit and implicit lexicon anchoring. According to official evaluation results, our submission for this group ranked 1st in the AMR subtask, 6th on PSD, and 7th on DM respectively, among 16 participating teams. For phrasal-anchoring, we proposed a CKY-based constituent tree parsing algorithm to resolve the anchor in UCCA, and our post-evaluation submission ranked 5th on UCCAsubtask.

3.1 Related Work and Anchoring Analysis

The 2019 Conference on Computational Language Learning (CoNLL) hosted a shared task on Cross-Framework Meaning Representation Parsing (MRP 2019, Oepen et al., 2019), which encourage participants in building a parser for five different meaning representations in three distinct flavors. Flavor-0 includes the DELPH-IN MRS Bi-lexical Dependencies (DM, Ivanova et al., 2012) and Prague Semantic Dependencies (PSD, Hajic et al., 2012; Miyao et al., 2014). Both frameworks under this representation have a syntactic backbone that is (either natively or by-proxy) based on bi-lexical dependency structures. As a result, the semantic concepts in these meaning representations can be anchored to the individual lexical units of the sentence. Flavor-1 includes Elementary Dependency Structures (EDS, Oepen and Lønning, 2006) and Universal Conceptual Cognitive Annotation framework (UCCA, Abend and Rappoport, 2013b), which shows an explicit, many-to-many anchoring of semantic concepts onto sub-strings of the underlying sentence. Finally, Flavor-2 includes Abstract Meaning Representation (AMR, Banarescu et al., 2013b), which is designed to abstract the meaning representation away from its surface token. But it leaves open the question of how these are derived.

3.1.1 Lexical-Anchoring Analysis on DM, PSD and AMR

Previous studies have shown that the nodes in AMR graphs are predominantly aligned with the surface lexical units, although explicit anchoring is absent from the AMR representation. In this section, we review the related work supporting the claim of the implicit

¹The code is available online at <https://github.com/utahnlp/lapa-mrp>

anchoring in AMR is actually lexical-anchoring, which can be merged into Flavor-0 when we consider the parsing methods on it.

3.1.1.1 Explicit Alignments: DM, PSD

[*introduction bi-lexical dependency*_{TODO}]

3.1.1.2 Implicit Anchoring in AMR

AMR tries to abstract the meaning representation away from the surface token. The absence of explicit anchoring can present difficulties for parsing. In this section, by extensive analysis on previous work AMR alignments, we show that AMR nodes can be implicitly aligned to the lexical tokens in a sentence.

AMR-to-String Alignments. A straightforward solution to find the missing anchoring in an AMR Graph is to align it with a sentence; We denote it as AMR-to-String alignment. ISI alignments Pourdamghani et al. (2014) first linearizes the AMR graph into a sequence, and then use IBM word alignment model Brown et al. (1993) to align the linearized sequence of concepts and relations with tokens in the sentence. According to the AMR annotation guidelines and error analysis of ISI aligner, some of the nodes or relations are evoked by subwords, e.g., the whole graph fragment (p/possible-01 :polarity -) is evoked by word "impossible", where the subword "im-" actually evoked the relation polarity and concept "-"; On the other side, sometimes concepts are evoked by multiple words, e.g., named entities, (c/city :name (n/name :op1 "New":op2 "York")), which also happens in explicit anchoring of DM and PSD. Hence, aligning and parsing with recategorized graph fragments are a natural solution in aligners and parsers. JAMR aligner Flanigan et al. (2014) uses a set of rules to greedily align single tokens, special entities and a set of multiple word expression to AMR graph fragments, which is widely used in previous AMR parsers (e.g. Flanigan et al., 2014; Wang et al., 2015; Artzi et al., 2015; Pust et al., 2015; Peng et al., 2015; Konstas et al., 2017; Wang and Xue, 2017). Other AMR-to-String Alignments exists, such as the extended HMM-based aligner. To consider more structure info in the linearized AMR concepts, Wang and Xue (2017) proposed a Hidden Markov Model (HMM)-based alignment method with a novel graph distance. All of them report over 90% F-score on their own hand-aligned datasets, which shows that AMR-to-String

alignments are almost token-level anchoring.

AMR-to-Dependency Alignments. Chen and Palmer (2017) first tries to align an AMR graph with a syntactic dependency tree. Szubert et al. (2018) conducted further analysis on dependency tree and AMR interface. It showed 97% of AMR edges can be evoked by words or the syntactic dependency edges between words. Those nodes in the dependency graph are anchored to each lexical token in the original sentence. Hence, this observation indirectly shows that AMR nodes can be aligned to the lexical tokens in the sentence.

Both AMR-to-String and AMR-to-dependency alignments shows that AMR nodes, including recategorized AMR graph fragments, do have implicit lexical anchoring. Based on this, Lyu and Titov (2018) propose to treat token-node alignments as discrete and exclusive alignment matrix and learn the latent alignment jointly with parsing. Recently, attention-based seq2graph model also achieved the state-of-the-art accuracy on AMR parsing Zhang et al. (2019b). However, whether the attention weights can be explained as AMR alignments needs more investigation in future.

3.1.1.3 Lexical-Anchoring

According to the bi-lexical dependency structures of DM and PSD, and implicit lexical token anchoring on AMR, the nodes/categorized graph fragments of DM, PSD, and AMR are anchored to surface lexical units in an explicit or implicit way. Especially, those lexical units do not overlap with each other, and most of them are just single tokens, multiple word expression, or named entities. In other words, when parsing a sentence into DM, PSD, AMR graphs, tokens in the original sentence can be merged by looking up a lexicon dict when preprocessing and then may be considered as a single token for aligning or parsing.

3.1.2 Analysis on Phrasal Anchoring in UCCA and TOP

As a result, the semantic concepts in these meaning representations can be anchored to the individual lexical units of the sentence. Flavor-1 includes Elementary Dependency Structures (EDS, Oepen and Lønning, 2006) and Universal Conceptual Cognitive Annotation framework (UCCA, Abend and Rappoport, 2013b), which shows an explicit, many-to-many anchoring of semantic concepts onto sub-strings of the underlying sentence.

However, different from the lexical anchoring without overlapping, nodes in EDS and UCCA may align to larger overlapped word spans which involves syntactic or semantic phrasal structure. Nodes in UCCA do not have node labels or node properties, but all the nodes are anchored to the spans of the underlying sentence. Furthermore, the nodes in UCCA are linked into a hierarchical structure, with edges going between parent and child nodes. With certain exceptions (e.g. remote edges), the majority of the UCCA graphs are tree-like structures. According to the position as well as the anchoring style, nodes in UCCA can be classified into the following two types:

1. **Terminal nodes** are the leaf semantic concepts anchored to individual lexical units in the sentence

2. **Non-terminal nodes** are usually anchored to a span with more than one lexical units, thus usually overlapped with the anchoring of terminal nodes.

The similar classification of anchoring nodes also applies to the nodes in EDS, although they do not regularly form a recursive tree like UCCA. As the running example in Figure ??, most of the nodes belongs to terminal nodes, which can be explicitly anchored to a single token in the original sentence. However, those bold non-terminal nodes are anchored to a large span of words. For example, the node "undef_q" with span <53:100> is aligned to the whole substring starting from "other crops" to the end; The abstract node with label `imp_conj` are corresponding to the whole coordinate structure between soybeans and rice

3.1.3 Summary of Anchoring Analysis

In summary, by treating AMR as an implicitly lexically anchored MR, we propose a simplified taxonomy for parsing the five meaning representations.

Lexical-anchoring: DM, PSD, AMR. In Section 3.2, we proposed an unified graph-based parsing framework for lexical-anchoring, which use a latent alignment model to support both the explicit and implicit alignments.

Phrasal-anchoring: UCCA, TOP. In Section 3.3, we proposed an unified span-based CKY parsing for both UCCA and TOP.

3.2 Latent Alignment Model for Lexical-Anchoring Graph-based Parsing

In this section, we will introduce the two-stage framework for parsing the DM, PSD, and AMR graphs §3.2.1. Then we resolve the alignment problem with a latent alignment model §??

3.2.1 Two-stage Graph-Based Parsing

Before formulating the graph-based model into a probabilistic model as Equation 3.1, we denote some notations: C, R are sets of concepts (nodes) and relations (edges) in the graph, and w is a sequence of tokens. $a \in \mathbb{Z}^m$ as the alignment matrix, each a_i is the index of aligned token where i th node aligned to. When modeling the negative log likelihood loss (NLL), with independence assumption between each node and edge, we decompose it into node- and edge-identification pipelines.

$$\begin{aligned}
 & NLL(P(C, R \mid w)) \\
 &= -\log(P(C, R \mid w)) \\
 &= -\log\left(\sum_a P(a) P(C, R \mid w, a)\right) \\
 &= -\log\left(\sum_a P(a) P(R \mid w, a, c) P(c \mid w, a)\right) \\
 &= -\log\left(\sum_a P(a) \prod_i^m P(c_i \mid h_{a_i}) \cdot \prod_{i,j=1}^m P(r_{ij} \mid h_{a_i}, c_i, h_{a_j}, c_j)\right)
 \end{aligned} \tag{3.1}$$

In DM, PSD, and AMR, every token will only be aligned once. Hence, we train a joint model to maximize the above probability for both node identification $P(c_i \mid h_{a_i})$ §?? and edge identification $P(r_{ij} \mid h_{a_i, c_i, h_{a_j}, c_j})$ §??. The alignment information is mainly used for training. We need to marginalize out the discrete alignment variable a to jointly learning the parameters in node identification and relation identification networks. We will introduce the latent alignment model in Section 3.2.2. Figure 3.1 summerize the unified two-stage graph based parsing framework. In the following subsections, we will explain the framework in more details.

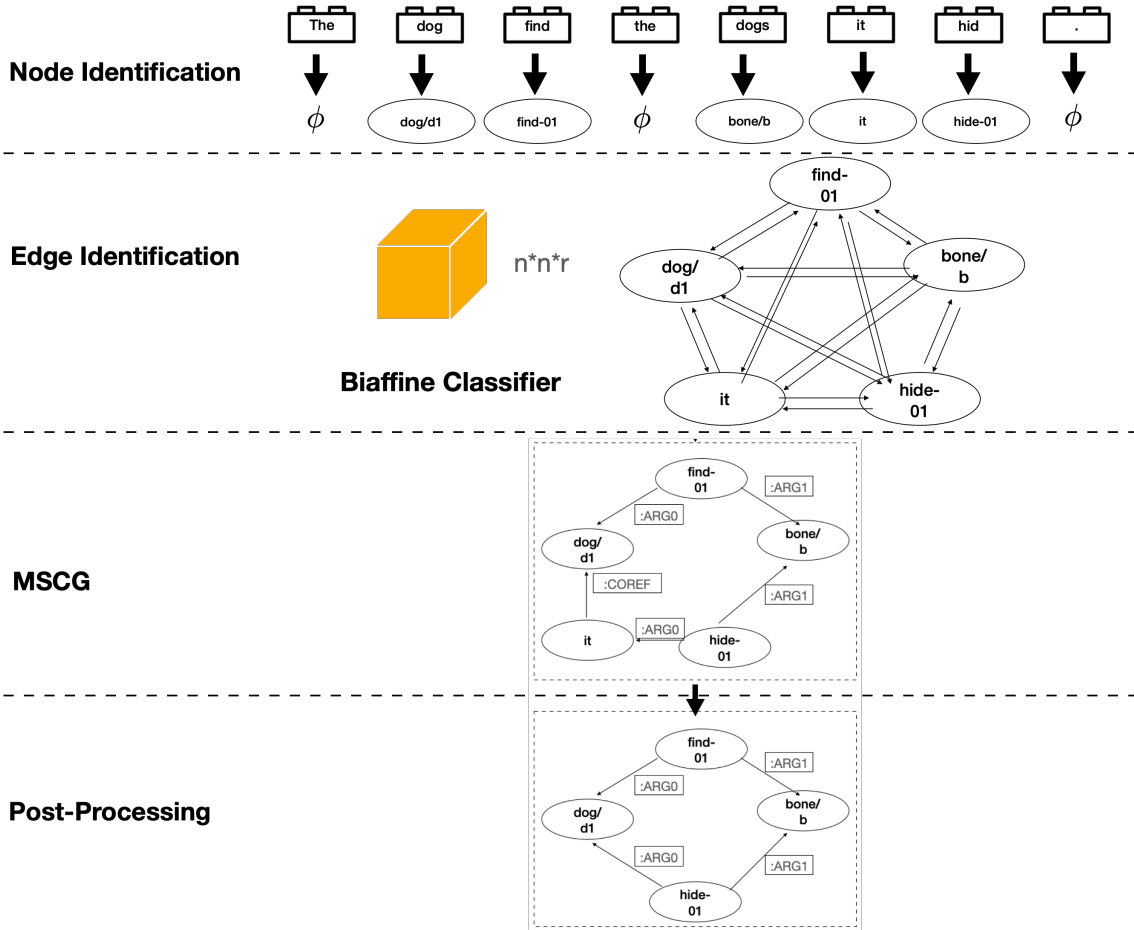


Figure 3.1: Architecture of graph-based model and inference, for running example [wsj#0209013]

3.2.1.1 Node Identification

Node Identification predicts a concept c given a word. A concept can be either *NULL* (when there is no semantic node anchoring to that word, e.g., the word is dropped), or a node label (e.g., lemma, sense, POS, name value in AMR, frame value in PSD), or other node properties. One challenge in node identification is the data sparsity issue. Many of the labels are from open sets derived from the input token, e.g., its lemma. Moreover, some labels are constrained by a deterministic label set given the word. Hence, we designed a copy mechanism (Luong et al., 2014) in our neural network architecture to decide whether to copying deterministic label given a word or estimate a classification probability from a fixed label set.

3.2.1.2 Edge Identification

By assuming the independence of each edge, we model the edges probabilities independently. Given two nodes and their underlying tokens, we predict the edge label as the semantic relation between the two concepts with a bi-affine classifier Dozat and Manning (2016).

3.2.1.3 Inference

In our two-stage graph-based parsing, after nodes are identified, edge identification only output a probability distribution over all the relations between identified nodes. However, we need to an inference algorithm to search for the *maximum spanning connected graph* from all the relations. We use Flanigan et al. (MSCG, 2014) to greedily select the most valuable edges from the identified nodes and their relations connecting them. As shown in Figure 3.1, an input sentence goes through preprocessing, node identification, edge identification, root identification, and MCSG to generate a final connected graph as structured output.

3.2.2 Latent Alignment Model

As the two-stage probabilistic model shown in ??, we need to marginalize all the alignment information a to learn the above two-stage neural networks for node and edge identification. We do the following computing for explicit and implicit alignments respectively.

Explicit Alignments. For DM, PSD, with explicit alignments a^* , we can simply use $P(a^*) = 1.0$ and other alignments $P(a|a \neq a^*) = 0.0$. In this case, with known alignment information, we don't need to worry the marginalization problem.

Implicit Alignments. However, For AMR, without gold alignments, one requires to compute all the valid alignments and then condition the node- and edge-identification methods on the alignments. However, it is computationally intractable to enumerate all combinatorial values for the discrete alignment variable. Hence, we estimate the latent alignments via variational inference, which has been initially used in Lyu and Titov (2018). In the following section, we firstly introduce the details of latent alignment model via continuous relaxation §3.2.2.1 and then we describe the details of variational inference §3.2.2.2, and we propose a error fix for computing KL-divergence with implicit Gumbel-Sinkhorn

distribution.

$$\begin{aligned}
& NLL(P(C, R \mid w)) \\
&= -\log \left(\sum_a P(a) \prod_i^m P(c_i \mid h_{a_i}) \right. \\
&\quad \left. \cdot \prod_{i,j=1}^m P(r_{ij} \mid h_{a_i}, c_i, h_{a_j}, c_j) \right)
\end{aligned} \tag{3.2}$$

3.2.2.1 Continuous Relaxation for Discrete Alignments

$$\begin{aligned}
& \log(P(C, R \mid w)) \geq \\
& E_Q[\log(P_\theta(c \mid w, a) P_\Phi(R \mid w, a, c))] \\
& - D_{KL}(Q_\Psi(a \mid c, R, w) \parallel P(a))
\end{aligned} \tag{3.3}$$

3.2.2.2 VAE, Perturb-and-Map, Gumble Sinkhorn

- Applying variational inference to reduce it into Evidence Lower Bound (ELBO, Kingma and Welling, 2013)
- The denominator Z_Ψ in Q can be estimated by Perturb-and-Max(MAP) Papandreou and Yuille (2011)

$$Q_\Psi(a \mid c, R, w) = \frac{\exp(\sum_{i=1}^n \phi(g_i, h_{a_i}))}{Z_\Psi(c, w)} \tag{3.4}$$

Where $\phi(g_i, h_{a_i})$ score each alignment link between node i and the corresponding words, g_i is node encoding, and h_{a_i} is encoding for the aligned token.

- Discrete argmax of a permutation can be estimated by Gumbel-Softmax Sinkhorn Networks Mena et al. (2018); Lyu and Titov (2018)

3.3 Minimal Span-based CKY Parsing Framework

Let us now see our phrasal-anchoring parser for UCCA and TOP. We introduce the transformation we used to reduce UCCA and TOP parsing into a unified constituent tree parsing task, and finally introduce the detailed CKY model for the constituent parsing.

3.3.1 Graph to Constituent Tree Transformation

UCCA to Consistent Tree. We propose to transform a graph into a constituent tree structure for parsing, which is also used in recent work Jiang et al. (2019). Figure 3.2 shows an example of transforming a UCCA graph into a constituent tree. The primary transformation assigns the original label of an edge to its child node. Then to make it compatible with parsers for standard PennTree Bank format, we add some auxiliary nodes such as special non-terminal nodes, TOP, HEAD, and special terminal nodes TOKEN and MWE. We remove all the “remote” annotation in UCCA since the constituent tree structure does not support reentrance. A fully compatible transformation should support both graph-to-tree and tree-to-graph transformation.

In our case, due to time constraints, we remove those remote edges and reentrance edges during training. Besides that, we also noticed that for multi-word expressions, the children of a parent node might not be in a continuous span (i.e., discontinuous constituent), which is also not supported by our constituent tree parser. Hence, when training the tree parser, by reattaching the discontinuous tokens to its nearest continuous parent nodes, we force every sub span are continuous in the transformed trees. We leave the postprocessing to recover those discontinuous as future work.

For inference, given an input sentence, we first use the trained constituent tree parsing model to parse it into a tree, and then we transform a tree back into a directed graph by assigning the edge label as its child’s node label, and deleting those auxiliary labels, adding anchors to every remaining node.

TOP to Consistent Tree. For the hierarchical dialog representation TOP, we also can transform it to a constituent tree structure. Figure 3.3 shows the transformation process for the utterance “Driving directions to the Eagles game”. In TOP tree shown in the up side of the figure, the leaf nodes are not single words as in constituent tree, and there are no other non-terminal nodes (such as part-of-speech tags) other than the intents and slots nodes. Hence, we decompose the original terminal nodes in TOP into separate tokens, and add a special parent node as TOK to each of the terminal token node. Finally, it forms the constituent tree as shown in the bottom.

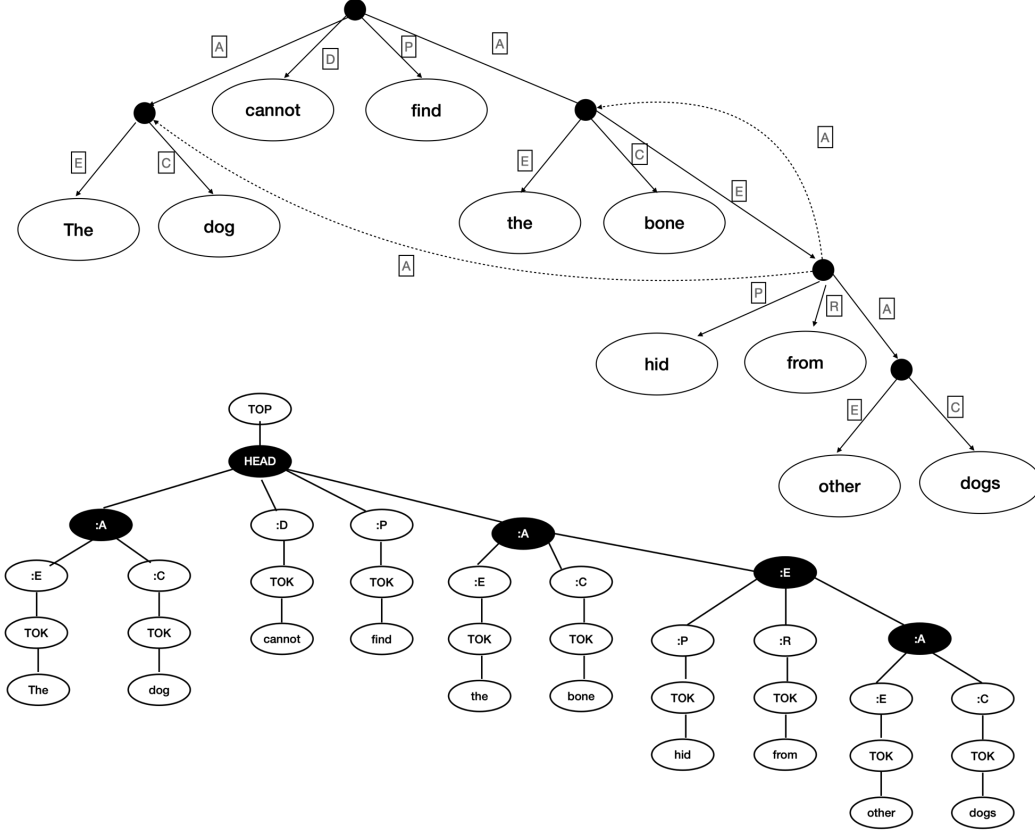


Figure 3.2: UCCA to Constituent Tree Transformation for [wsj#0209013]

3.3.2 A Unified Span-based Model for CKY Parsing

After transforming the UCCA graph into a constituent tree, we reduce the UCCA parsing into a constituent tree parsing problem. Similar to the previous work on UCCA constituent tree parsing Jiang et al. (2019), we use a minimal span-based CKY parser for constituent tree parsing. The intuition is to use dynamic programming to recursively split the span of a sentence recursively, as shown in Figure 3.2. The entire sentence can be splitted from top to bottom until each span is a single unsplittable tokens. For each node, we also need to assign a label. Two simplified assumptions are made when predicting the hole tree given a sentence. However, different with previous work, we use 8-layers with 8 heads transformer encoder, which shows better performance than LSTM in Kitaev and Klein (2018).

3.3.2.0.1 Tree Factorization In the graph-to-tree transformation, we move the edge label to its child node. By assuming the labels for each node are independent, we factorize the tree structure prediction as independent span-label prediction as Equation 3.5. How-

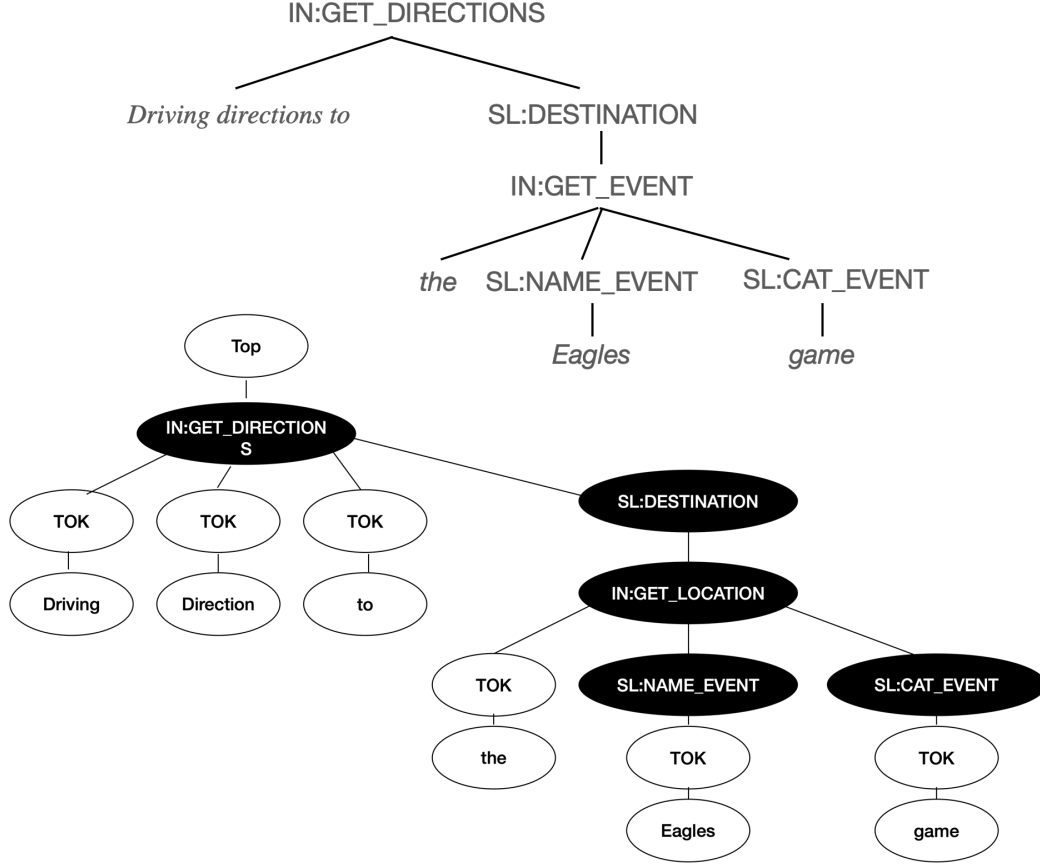


Figure 3.3: TOP to Constituent Tree Transformation for the utterance “Driving directions to the Eagles game”

ever, this assumption does not hold for UCCA. Please see more error analysis in §3.4.5

$$\begin{aligned}
 T^* &= \arg \max_T s(T) \\
 s(T) &= \sum_{(i,j,l) \in T} s(i,j,l)
 \end{aligned} \tag{3.5}$$

3.3.2.0.2 CKY Parsing By assuming the label prediction is independent of the splitting point, we can further factorize the whole tree as the following dynamic programming in Equation 3.6.

$$\begin{aligned}
 s_{\text{best}}(i, i+1) &= \max_l s(i, i+1, l) \\
 s_{\text{best}}(i, j) &= \max_l s(i, j, l) \\
 &\quad + \max_k [s_{\text{best}}(i, k) + s_{\text{best}}(k, j)]
 \end{aligned} \tag{3.6}$$

3.3.3 Span Encoding

For each span (i, j) , we represent the span encoding vector $v_{(i,j)} = [\vec{y}_j - \vec{y}_i] \oplus [y_{j+1}^{\leftarrow} - y_{i+1}^{\leftarrow}]$. \oplus denotes vector concatenation. Assuming a bidirectional sentence encoder, we use the forward and backward encodings \vec{y}_i and \vec{y}_i^{\leftarrow} of i_{th} word. Following the previous work, and we also use the loss augmented inference training. More details about the network architecture are in the Section 3.4.3

3.4 Experiments and Results

3.4.1 Dataset and Evaluation

For DM, PSD, EDS, we split the training set by taking WSJ section (00-19) as training, and section 20 as dev set. For other datasets, when developing and parameter tuning, we use splits with a ratio of 25:1:1. In our submitted model, we did not use multitask learning for training. Following the unified MRP metrics in the shared tasks, we train our model based on the development set and finally evaluate on the private test set. For more details of the metrics, please refer to the summarization of the MRP 2019 task Oepen et al. (2019),

	Lexicon Anchoring		
	DM	PSD	AMR
Top	1	≥ 1 (11.56%)	1
Node Label	Lemma	Lemma(*)	Lemma(*) + NeType(143+)
Node Properties	POS	POS	constant values
	semi(160*)_args(25)	wordid_sense(25)	polarity, Named entity
Edge Label	(45)	(91)	(94+)
Edge Properties	N/A	N/A	N/A
Connectivity	True	True	True
Training Data	35656	35656	57885
Test Data	3269	3269	1998

Table 3.1: Detailed classifiers in our model, round bracket means the number of output classes of our classify, * means copy mechanism is used in our classifier. At the end of shared task, EDS are not fully supported to get an official results, we leave it as our future work.

3.4.2 Summary of Implementation

We summarize our implementation for five meaning representations as Table ???. As we mentioned in the previous sections, we use latent-alignment graph-based parsing for lexical anchoring MRs (DM, PSD, AMR), and use CKY-based constituent parsing phrasal

	Phrase Anchoring	
	EDS	UCCA
Top	1	1
Node Label	_lemma(*)_semi_sense	N/A
Node Properties		N/A
	carg: constant value	N/A
Edge Label	(45)	(15)
Edge Properties	N/A	"remote"
Connectivity	True	True
Training Data	35656	6485
Test Data	3269	1131

Table 3.2: Detailed classifiers in our model, round bracket means the number of output classes of our classifier, * means copy mechanism is used in our classifier. At the end of shared task, EDS are not fully supported to get an official results, we leave it as our future work.

anchoring in MRs (UCCA, EDS). This section gives information about various decision for our models.

3.4.2.1 Top

The first row "Top" shows the numbers of root nodes in the graph. We can see that for PSD, 11.56% of graphs with more than 1 top nodes. In our system, we only predict one top node with a N (N is size of identified nodes) way classifier, and then fix this with a post-processing strategy. When our model predicts one node as the top node, and if we find additional coordination nodes with it, we add the coordination node also as the top node.

3.4.2.2 Node

Except for UCCA, all other four MRs have labeled nodes, the row "Node Label" shows the templates of a node label. For DM and PSD, the node label is usually the lemma of its underlying token. But the lemma is neither the same as one in the given companion data nor the predicted by Stanford Lemma Annotators. One common challenge for predicting the node labels is the open label set problem. Usually, the lemma is one of the morphology derivations of the original word. But the derivation rule is not easy to create manually. In our experiment, we found that handcrafted rules for lemma prediction only works worse than classification with copy mechanism, except for DM.

For AMR and EDS, there are other components in the node labels beyond the lemma.

Especially, the node label for AMR also contains more than 143 fine-grained named entity types; for EDS, it uses the full SEM-I entry as its node label, which requires extra classifiers for predicting the corresponding sense. In addition to the node label, the properties of the label also need to be predicted. Among them, node properties of DM are from the SEMI sense and arguments handler, while for PSD, senses are constrained the senses in the predefined the vallex lexicon.

3.4.2.3 Edge

Edge predication is another challenge in our task because of its large label set (from 45 to 94) as shown in row “Edge Label”, the round bracket means the number of output classes of our classifiers. For Lexical anchoring MRs, edges are usually connected between two tokens, while phrasal anchoring needs extra effort to figure out the corresponding span with that node. For example, in UCCA parsing, To predict edge labels, we first predicted the node spans, and then node labels based that span, and finally we transform back the node label into edge label.

3.4.2.4 Connectivity

Beside the local label classification for nodes and edges, there are other global structure constraints for all five MRs: All the nodes and edges should eventually form a connected graph. For lexical anchoring, we use MSCG algorithm to find the maximum connected graph greedily; For phrasal anchoring, we use dynamic programming to decoding the constituent tree then deterministically transforming back to a connected UCCA Graph ²

3.4.3 Model Setup

For lexical-anchoring model setup, our network mainly consists of node and edge prediction model. For AMR, DM, and PSD, they all use one layer Bi-directional LSTM for input sentence encoder, and two layers Bi-directional LSTM for head or dependent node encoder in the bi-affine classifier. For every sentence encoder, it takes a sequence of word embedding as input (We use 300 dimension Glove here), and then their output will pass a softmax layer to predicting output distribution. For the latent AMR model, to

²Due to time constraint, we ignored all the discontinuous span and remote edges in UCCA

model the posterior alignment, we use another Bi-LSTM for node sequence encoding. For phrasal-anchoring model setup, we follow the original model set up in Kitaev and Klein (2018), and we use 8-layers 8-headers transformer with position encoding to encode the input sentence.

For all sentence encoders, we also use the character-level CNN model as character-level embedding without any pre-trained deep contextualized embedding model. Equipping our model with Bert or multi-task learning is promising to get further improvement. We leave this as our future work.

Our models are trained with Adam Kingma and Ba (2015), using a batch size 64 for a graph-based model, and 250 for CKY-based model. Hyper-parameters were tuned on the development set, based on labeled F1 between two graphs. We exploit early-stopping to avoid over-fitting.

3.4.4 Results

At the time of official evaluation, we submitted three lexical anchoring parser, and then we submitted another phrasal-anchoring model for UCCA parsing during post-evaluation stage, and we leave EDS parsing as future work. The following sections are the official results and error breakdowns for lexical-anchoring and phrasal-anchoring respectively.

3.4.4.0.1 Official Results on Lexical Anchoring Table 3.3 shows the official results for our lexical-anchoring models on AMR, DM, PSD. By using our latent alignment based AMR parser, our system ranked top 1 in the AMR subtask, and outperformed the top 5 models in large margin. Our parser on PSD ranked 6, but only 0.02% worse than the top 5 model. However, official results on DM and PSD shows that there is still around 2.5 points performance gap between our model and the top 1 model.

MR	Ours (P/R/F1)	Top 1/3/5 (F1)
AMR(1)	75/71/73.38	73.38/71.97/71.72
PSD(6)	89/89/ 88.75	90.76/89.91/ 88.77
DM(7)	93/92/92.14	94.76/94.32/93.74

Table 3.3: Official results overview on unified MRP metric, we selected the performance from top 1/3/5 system(s) for comparison

3.4.4.0.2 Official Results on Phrasal Anchoring Table 3.4 shows that our span-based CKY model for UCCA can achieve 74.00 F1 score on official test set, and ranked 5th. When

adding ELMo Peters et al. (2018a) into our model, it can further improve almost 3 points on it.

MR	Ours (P/R/F1)	Top 1/3/5 (F1)
UCCA(5)	80.83/73.42/ 76.94	81.67/77.80/73.22
EDS	N/A	94.47/90.75/89.10

Table 3.4: Official results overview on unified MRP metric, we selected the performance from top 1/3/5 system(s) for comparison. It shows our UCCA model for post-evaluation can rank 5th

3.4.5 Error Breakdown

Table 3.5, 3.6, 3.7 and 3.8 shows the detailed error breakdown of AMR, DM, PSD and UCCA respectively. Each column in the table shows the F1 score of each subcomponent in a graph: top nodes, node labels, node properties, node anchors, edge labels, and overall F1 score. No anchors for AMR, and no node label and properties for UCCA. We show the results of MRP metric on two datasets. “all” denotes all the examples for that specific MR, while lpps are a set of 100 sentences from *The Little Prince*, and annotated in all five meaning representations. To better understand the performance, we also reported the official results from two baseline models TUPA Hershcovich and Arviv (2019) and ERG Oepen and Flickinger (2019).

		data	tops	labels	prop	edges	all
TUPA	all	63.95	57.20	22.31	36.41	44.73	
single	lpps	71.96	55.52	26.42	36.38	47.04	
TUPA	all	61.30	39.80	27.70	27.35	33.75	
multi	lpps	72.63	50.11	20.25	33.12	43.38	
Ours(1)	all	<u>65.92</u>	82.86	77.26	63.57	73.38	
	lpps	<u>72.00</u>	78.71	58.93	63.96	71.11	
Top 2	all	78.15	82.51	71.33	63.21	72.94	
	lpps	83.00	76.24	51.79	60.43	69.03	

Table 3.5: Our parser on AMR ranked 1st. This table shows the error breakdown when comparing to the baseline TUPA model and top 2 Che et al. (2019) in official results

3.4.5.1 Error Analysis on Lexical-Anchoring

As shown in Table 3.5, our AMR parser is good at predicting node properties and consistently perform better than other models in all subcomponent, except for top prediction.

		data	tops	labels	prop	anchors	edges	all
ERG	all	91.83	98.22	95.25	98.82	90.76	95.65	
	lpps	95.00	97.32	97.75	99.46	92.71	97.03	
Top 1	all	93.23	94.14	94.83	98.40	91.55	94.76	
	lpps	96.48	91.85	94.36	99.04	93.28	94.64	
Ours(7)	all	<u>70.95</u>	93.96	92.13	97.25	86.45	92.14	
	lpps	<u>84.00</u>	90.55	91.91	97.96	87.24	91.82	

Table 3.6: Our parser on DM ranked 7th. This table shows the error breakdown when comparing to the model ranked Top 1 Li et al. (2019) in official results

		data	tops	labels	prop	anchors	edges	all
Top 1	all	93.45	94.68	91.78	98.35	77.79	90.76	
	lpps	93.33	91.73	84.37	98.40	77.63	88.34	
Ours(6)	all	<u>82.01</u>	94.18	91.28	96.94	72.40	88.75	
	lpps	<u>85.85</u>	90.48	82.63	95.97	73.60	85.83	

Table 3.7: Our parser on PSD ranked 6th. This table shows the error breakdown when comparing to the model ranked top 1 Donatelli et al. (2019) in official results

Node properties in AMR are usually named entities, negation, and some other quantity entities. In our system, we recategorize the graph fragments into a single node, which helps for both alignments and structured inference for those special graph fragments. We see that all our 3 models perform almost as good as the top 1 model of each subtask on node label prediction, but they perform worse on top and edge prediction. It indicates that our bi-affine relation classifier are main bottleneck to improve. Moreover, we found the performance gap between node labels and node anchors are almost consistent, it indicates that improving our model on predicting NULL nodes may further improve node label prediction as well. Moreover, we believe that multi-task learning and pre-trained deep models such as BERT Devlin et al. (2018) may also boost the performance of our parser in future.

3.4.5.2 Error Analysis on Phrasal-Anchoring

According to Table 3.8, our model with ELMo works slightly better than the top 1 model on anchors prediction. It means our model is good at predicting the nodes in UCCA and we believe that it is also helpful for prediction phrasal anchoring nodes in EDS.

However, when predicting the edge and edge attributes, our model performs 7-8 points worse than the top 1 model. In UCCA, an edge label means the relation between a parent nodes and its children. In our UCCA transformation, we assign edge label as the node

		data	tops	anchors	edge	attr	all
TUPA single	all	78.73	69.17	16.96	15.18	27.56	
	lpps	86.03	76.26	28.32	24.00	40.06	
TUPA multi	all	84.92	65.74	12.99	9.07	23.65	
	lpps	88.89	77.76	26.45	18.32	41.04	
Che et al. (2019)	all	1.00	95.36	72.66	61.98	81.67	
	lpps	1.00	96.99	73.08	48.37	82.61	
Ours(*5)	all	98.85	94.92	60.17	0.00	74.00	
	lpps	96.00	96.75	60.20	0.00	75.17	
Ours + ELMo	all	99.38	95.70	64.88	0.00	76.94	
	lpps	98.00	96.84	66.63	0.00	78.77	

Table 3.8: Our UCCA parser in post-evaluation ranked 5th according to the original official evaluation results. This table shows the error breakdown when comparing to the model ranked top 1 Che et al. (2019) in official results. * denotes the ranking of post-evaluation results

label of its child and then predict with only child span encoding. Thus it actually misses important information from the parent node. Hence, in future, more improvement can be done to use both child and parent span encoding for label prediction, or even using another span-based bi-affine classifier for edge prediction, or remote edge recovering.

3.5 Chapter Summary

In summary, by analyzing the AMR alignments, we show that implicit AMR anchoring is actually lexical-anchoring based. Thus we propose to regroup five meaning representations as two groups: lexical-anchoring and phrasal-anchoring. For lexical anchoring, we suggest to parse DM, PSD, and AMR in a unified latent-alignment based parsing framework. Our submission ranked top 1 in AMR sub-task, ranked 6th and 7th in PSD and DM tasks. For phrasal anchoring, by reducing UCCA graph into a constituent tree-like structure, and then use the span-based CKY parsing to parse their tree structure, our method would rank 5th in the original official evaluation results.

CHAPTER 4

MODELING ON SENTENCE-ANCHORING FOR DIALOG IN THERAPY

Conversational agents have long been studied in the context of psychotherapy, going back to chatbots such as ELIZA Weizenbaum (1966) and PARRY Colby (1975). Research in modeling such dialogue has largely sought to simulate a participant in the conversation.

In this paper, we argue for modeling dialogue *observers* instead of participants, and focus on psychotherapy. An observer could help an *ongoing* therapy session in several ways. First, by monitoring fidelity to therapy standards, a helper could guide both veteran and novice therapists towards better patient outcomes. Second, rather than generating therapist utterances, it could suggest the type of response that is appropriate. Third, it could alert a therapist about potentially important cues from a patient. Such assistance would be especially helpful in the increasingly prevalent online or text-based counseling services.¹

We ground our study in a style of therapy called Motivational Interviewing (MI, Miller and Rollnick, 2003, 2012), which is widely used for treating addiction-related problems. To help train therapists, and also to monitor therapy quality, utterances in sessions are annotated using a set of behavioral codes called Motivational Interviewing Skill Codes (MISC, Miller et al., 2003). Table 4.1 shows standard therapist and patient (*i.e.*, client) codes with examples. Recent NLP work (Tanana et al., 2016; Xiao et al., 2016; Pérez-Rosas et al., 2017; Huang et al., 2018, *inter alia*) has studied the problem of using MISC to assess *completed* sessions. Despite its usefulness, automated post hoc MISC labeling does not address the desiderata for ongoing sessions identified above; such models use information from utterances yet to be said. To provide real-time feedback to therapists, we define two

¹For example, Crisis Text Line (<https://www.crisistextline.org>), 7 Cups (<https://www.7cups.com>), etc.

Code	Count	Description	Examples
Client Behavioral Codes			
FN	47715	Follow/ Neutral: unrelated to changing or sustaining behavior.	"You know, I didn't smoke for a while." "I have smoked for forty years now."
CT	5099	Utterances about changing unhealthy behavior.	"I want to stop smoking."
ST	4378	Utterances about sustaining unhealthy behavior.	"I really don't think I smoke too much."
Therapist Behavioral Codes			
FA	17468	Facilitate conversation	"Mm Hmm.", "OK.", "Tell me more."
GI	15271	Give information or feedback.	"I'm Steve.", "Yes, alcohol is a depressant."
RES	6246	Simple reflection about the client's most recent utterance.	C: "I didn't smoke last week" T: "Cool, you avoided smoking last week."
REC	4651	Complex reflection based on a client's conversation history.	C: "I didn't smoke last week." T: "You mean things begin to change".
QUC	5218	Closed question	"Did you smoke this week?"
QUO	4509	Open question	"Tell me more about your week."
MIA	3869	MI adherent, <i>e.g.</i> , affirmation, advising with permission, etc.	"You've accomplished a difficult task." "Is it OK if I suggested something?"
MIN	1019	MI non-adherent, <i>e.g.</i> , confront, advising without permission, etc.	"You hurt the baby's health for cigarettes?" "You ask them not to drink at your house."

Table 4.1: Distribution, description and examples of MISC labels.

complementary dialogue observers:

1. **Categorization:** Monitoring an ongoing session by predicting MISC labels for therapist and client utterances as they are made.
2. **Forecasting:** Given a dialogue history, forecasting the MISC label for the next utterance, thereby both alerting or guiding therapists.

Via these tasks, we envision a helper that offers assistance to a therapist in the form of MISC labels.

We study modeling challenges associated with these tasks related to: (1) representing words and utterances in therapy dialogue, (2) ascertaining relevant aspects of utterances and the dialogue history, and (3) handling label imbalance (as evidenced in Table 4.1). We develop neural models that address these challenges in this domain.

Experiments show that our proposed models outperform baselines by a large margin. For the categorization task, our models even outperform previous session-informed approaches that use information from future utterances. For the more difficult forecasting task, we show that even without having access to an utterance, the dialogue history pro-

vides information about its MISC label. We also report the results of an ablation study that shows the impact of the various design choices.²

In summary, in this chapter, we (1) define the tasks of categorizing and forecasting Motivational Interviewing Skill Codes to provide real-time assistance to therapists, (2) propose neural models for both tasks that outperform several baselines, and (3) show the impact of various modeling choices via extensive analysis.

4.1 Background and Motivation

Motivational Interviewing (MI) is a style of psychotherapy that seeks to resolve a client’s ambivalence towards their problems, thereby motivating behavior change. Several meta-analyses and empirical studies have shown the high efficacy and success of MI in psychotherapy Burke et al. (2004); Martins and McNeil (2009); Lundahl et al. (2010). However, MI skills take practice to master and require ongoing coaching and feedback to sustain Schwalbe et al. (2014). Given the emphasis on using specific types of linguistic behaviors in MI (*e.g.*, open questions and reflections), fine-grained behavioral coding plays an important role in MI theory and training.

Motivational Interviewing Skill Codes (MISC, table 4.1) is a framework for coding MI sessions. It facilitates evaluating therapy sessions via utterance-level labels that are akin to dialogue acts Stolcke et al. (2000); Jurafsky and Martin (2019), and are designed to examine therapist and client behavior in a therapy session.³

As Table 4.1 shows, client labels mark utterances as discussing changing or sustaining problematic behavior (**CT** and **ST**, respectively) or being neutral (**FN**). Therapist utterances are grouped into eight labels, some of which (**RES**, **REC**) correlate with improved outcomes, while MI non-adherent (**MIN**) utterances are to be avoided. MISC labeling was originally done by trained annotators performing multiple passes over a session recording or a transcript. Recent NLP work speeds up this process by automatically annotating a completed MI session (*e.g.*, Tanana et al., 2016; Xiao et al., 2016; Pérez-Rosas et al., 2017).

²The code is available online at <https://github.com/utahnlp/therapist-observer>.

³The original MISC description of Miller et al. (2003) included 28 labels (9 client, 19 therapist). Due to data scarcity and label confusion, various strategies are proposed to merge the labels into a coarser set. We adopt the grouping proposed by Xiao et al. (2016); the appendix gives more details.

i	s_i	u_i	l_i
1	T:	Have you used drugs recently?	QUC
2	C:	I stopped for a year, but relapsed.	FN
3	T:	You will suffer if you keep using.	MIN
4	C:	Sorry, I just want to quit.	CT
...

Table 4.2: An example of ongoing therapy session

Instead of providing feedback to a therapist after the completion of a session, can a dialogue observer provide online feedback? While past work has shown the helpfulness of post hoc evaluations of a session, prompt feedback would be more helpful, especially for MI non-adherent responses. Such feedback opens up the possibility of the dialogue observer influencing the therapy session. It could serve as an assistant that offers suggestions to a therapist (novice or veteran) about how to respond to a client utterance. Moreover, it could help alert the therapist to potentially important cues from the client (specifically, CT or ST).

4.2 Task Definitions

In this section, we will formally define the two NLP tasks corresponding to the vision in §?? using the conversation in table 4.2 as a running example.

Suppose we have an ongoing MI session with utterances u_1, u_2, \dots, u_n : together, the dialogue history H_n . Each utterance u_i is associated with its speaker s_i , either C (client) or T (therapist). Each utterance is also associated with the MISC label l_i , which is the object of study. We will refer to the last utterance u_n as the *anchor*.

We will define two classification tasks over a fixed dialogue history with n elements — *categorization* and *forecasting*. As the conversation progresses, the history will be updated with a sliding window. Since the therapist and client codes share no overlap, we will design separate models for the two speakers, giving us four settings in all.

Task 1: Categorization. The goal of this task is to provide real-time feedback to a therapist during an ongoing MI session. In the running example, the therapist’s confrontational response in the third utterance is not MI adherent (MIN); an observer should flag it as such to bring the therapist back on track. The client’s response, however, shows an inclination to change their behavior (CT). Alerting a therapist (especially a novice) can help guide the

conversation in a direction that encourages it.

In essence, we have the following real-time classification task: *Given the dialogue history H_n which includes the speaker information, predict the MISC label l_n for the last utterance u_n .*

The key difference from previous work in predicting MISC labels is that we are restricting the input to the real-time setting. As a result, models can only use the dialogue history to predict the label, and in particular, we can not use models such as a conditional random field or a bi-directional LSTM that need both past and future inputs.

Task 2: Forecasting. A real-time therapy observer may be thought of as an expert therapist who guides a session with suggestions to the therapist. For example, after a client discloses their recent drug use relapse, a novice therapist may respond in a confrontational manner (which is not recommended, and hence coded **MIN**). On the other hand, a seasoned therapist may respond with a complex reflection (**REC**) such as “*Sounds like you really wanted to give up and you’re unhappy about the relapse.*” Such an expert may also anticipate important cues from the client.

The forecasting task seeks to mimic the intent of such a seasoned therapist: *Given a dialogue history H_n and the next speaker’s identity s_{n+1} , predict the MISC code l_{n+1} of the yet unknown next utterance u_{n+1} .*

The MISC forecasting task is a previously unstudied problem. We argue that forecasting the type of the next utterance, rather than selecting or generating its text as has been the focus of several recent lines of work (e.g., Schatzmann et al., 2005; Lowe et al., 2015b; Yoshino et al., 2018), allows the human in the loop (the therapist) the freedom to creatively participate in the conversation within the parameters defined by the seasoned observer, and perhaps even rejecting suggestions. Such an observer could be especially helpful for training therapists Imel et al. (2017). The forecasting task is also related to recent work on detecting antisocial comments in online conversations Zhang et al. (2018) whose goal is to provide an early warning for such events.

4.3 Models for MISC Prediction

Modeling the two tasks defined in §?? requires addressing four questions: (1) How do we encode a dialogue and its utterances? (2) Can we discover discriminative words in each utterance? (3) Can we discover which of the previous utterances are relevant? (4) How do

we handle label imbalance in our data? Many recent advances in neural networks can be seen as plug-and-play components. To facilitate the comparative study of models, we will describe components that address the above questions. In the rest of the paper, we will use **boldfaced** terms to denote vectors and matrices and SMALL CAPS to denote component names.

4.3.1 Encoding Dialogue

Since both our tasks are classification tasks over a dialogue history, our goal is to convert the sequence of utterances into a single vector that serves as input to the final classifier.

We will use a hierarchical recurrent encoder (Li et al., 2015; Sordoni et al., 2015; Serban et al., 2016, and others) to encode dialogues, specifically a hierarchical gated recurrent unit (HGRU) with an utterance and a dialogue encoder. We use a bidirectional GRU over word embeddings to encode utterances. As is standard, we represent an utterance u_i by concatenating the final forward and reverse hidden states. We will refer to this utterance vector as v_i . Also, we will use the hidden states of each word as inputs to the attention components in §4.3.2. We will refer to such contextual word encoding of the j^{th} word as v_{ij} . The dialogue encoder is a unidirectional GRU that operates on a concatenation of utterance vectors v_i and a trainable vector representing the speaker s_i .⁴ The final state of the GRU aggregates the entire dialogue history into a vector H_n .

The HGRU skeleton can be optionally augmented with the word and dialogue attention described next. All the models we will study are two-layer MLPs over the vector H_n that use a ReLU hidden layer and a softmax layer for the outputs.

4.3.2 Word-level Attention

Certain words in the utterance history are important to categorize or forecast MISC labels. The identification of these words may depend on the utterances in the dialogue. For example, to identify that an utterance is a simple reflection (**RES**) we may need to discover that the therapist is mirroring a recent client utterance; the example in table 4.1

⁴For the dialogue encoder, we use a unidirectional GRU because the dialogue is incomplete. For words, since the utterances are completed, we can use a BiGRU.

illustrates this. Word attention offers a natural mechanism for discovering such patterns.

We can unify a broad collection of attention mechanisms in NLP under a single high level architecture Galassi et al. (2019). We seek to define attention over the word encodings v_{ij} in the history (called queries), guided by the word encodings in the anchor v_{nk} (called keys). The output is a sequence of attention-weighted vectors, one for each word in the i^{th} utterance. The j^{th} output vector a_j is computed as a weighted sum of the keys:

$$a_{ij} = \sum_k \alpha_j^k v_{nk} \quad (4.1)$$

The weighting factor α_j^k is the attention weight between the j^{th} query and the k^{th} key, computed as

$$\alpha_j^k = \frac{\exp(f_m(v_{nk}, v_{ij}))}{\sum_{j'} \exp(f_m(v_{nk}, v_{ij'}))} \quad (4.2)$$

Here, f_m is a match scoring function between the corresponding words, and different choices give us different attention mechanisms.

Finally, a combining function f_c combines the original word encoding v_{ij} and the above attention-weighted word vector a_{ij} into a new vector representation z_{ij} as the final representation of the query word encoding:

$$z_{ij} = f_c(v_{ij}, a_{ij}) \quad (4.3)$$

The attention module, identified by the choice of the functions f_m and f_c , converts word encodings in each utterance v_{ij} into attended word encodings z_{ij} . To use them in the HGRU skeleton, we will encode them a second time using a BiGRU to produce attention-enhanced utterance vectors. For brevity, we will refer to these vectors as v_i for the utterance u_i . If word attention is used, these attended vectors will be treated as word encodings.

To complete this discussion, we need to instantiate the two functions. We use two commonly used attention mechanisms: BiDAF Seo et al. (2016) and gated matchLSTM Wang et al. (2017). For simplicity, we replace the sequence encoder in the latter with a BiGRU and refer to it as GMGRU. Table 4.3 shows the corresponding definitions of f_c and f_m . We refer the reader to the original papers for further details. In subsequent sections, we will refer to the two attended versions of the HGRU as BiDAF^H and GMGRU^H.

Method	f_m	f_c
BiDAF	$v_{nk} v_{ij}^T$	$[v_{ij}; a_{ij};$ $v_{ij} \odot a_{ij}; v_{ij} \odot a']$
GMGRU	$w^e \tanh(W^k v_{nk}$ $+ W^q [v_{ij}; h_{j-1}])$	$[v_{ij}; a_{ij}]$

Table 4.3: Summary of word attention mechanisms. We simplify BiDAF with multiplicative attention between word pairs for f_m , while GMGRU uses additive attention influenced by the GRU hidden state. The vector $w_e \in \mathbb{R}^d$, and matrices $W^k \in \mathbb{R}^{d \times d}$ and $W^q \in \mathbb{R}^{2d \times 2d}$ are parameters of the BiGRU. The vector h_{j-1} is the hidden state from the BiGRU in GMGRU at previous position $j - 1$. For combination function, BiDAF concatenates bidirectional attention information from both the key-aware query vector a_{ij} and a similarly defined query-aware key vector a' . GMGRU uses simple concatenation for f_c .

4.3.3 Utterance-level Attention

While we assume that the history of utterances is available for both our tasks, not every utterance is relevant to decide a MISC label. For categorization, the relevance of an utterance to the anchor may be important. For example, a complex reflection (REC) may depend on the relationship of the current therapist utterance to one or more of the previous client utterances. For forecasting, since we do not have an utterance to label, several previous utterances may be relevant. For example, in the conversation in Table 4.2, both u_2 and u_4 may be used to forecast a complex reflection.

To model such utterance-level attention, we will employ the multi-head, multi-hop attention mechanism used in Transformer networks Vaswani et al. (2017). As before, due to space constraints, we refer the reader to the original work for details. We will use the (Q, K, V) notation from the original paper here. These matrices represent a query, key and value respectively. The multi-head attention is defined as:

$$\text{Multihead}(Q, K, V) = [\text{head}_1; \dots; \text{head}_h] W^O \quad (4.4)$$

$$\text{head}_i = \text{softmax} \left(\frac{QW_i^Q (KW_i^K)^T}{\sqrt{d_k}} \right) VW_i^V$$

The W_i 's refer to projection matrices for the three inputs, and the final W^O projects the concatenated heads into a single vector.

The choices of the query, key and value defines the attention mechanism. In our work, we compare two variants: *anchor-based attention*, and *self-attention*. The anchor-based attention is defined by $Q = [v_n]$ and $K = V = [v_1 \dots v_n]$. Self-attention is defined by

Skeleton	Annotating	Forecasting
CON	$\mathbf{v}_n^{seg}, \mathbf{v}_n^{wordatt}, \mathbf{v}_n$	\mathbf{C}_n
HGRU	$\mathbf{H}_n, \mathbf{v}_n^{selfatt}, \mathbf{v}_n$	\mathbf{H}_n

Table 4.4: Input options for annotating and forecasting tasks based on CON and HGRU skeletons.

setting all three matrices to $[\mathbf{v}_1 \cdots \mathbf{v}_n]$. For both settings, we use four heads and stacking them for two hops, and refer to them as SELF_{42} and ANCHOR_{42} .

4.3.4 Predicting and Training

Predicting. From up to the bottom, every component will produce some of useful representation for inferences in our tasks. The dialogue encoding vector \mathbf{H}_n , as the final of the unidirectional GRU, it is also the contextual utterance encoding \mathbf{h}_{u_n} of u_n . Hence \mathbf{H}_n can be directly used as a representaion of u_n for classification in annotating tasks, also can be used as a representation of whole dialogue for forecasting task. Hence in HGRU setting, we always use \mathbf{H}_n as the base option as input for inference.

However, for CON skeleton, the final state \mathbf{C}_n does not exactly represent the segment u_n in the whole concatenated dialogue. Hence, we concatenate the hidden state of the start position (0) and end position (T) of u_n into $\mathbf{v}_n^{seg} = [\mathbf{h}_{u_n^0}; \mathbf{h}_{u_n^T}]$, which is contextual utterance encoding in CON mode.

Beside the above \mathbf{H}_n and \mathbf{C}_n contextual utterance encoding in dialogue level, our components also produced the original utterance encoding \mathbf{v}_n from utterance encoder. Whats'more, in **CON** mode, we can use history-aware utterance encoding $\mathbf{v}_n^{wordatt}$. While in HGRU, it produced a self-attentive utterance encoding. We denote it as $\mathbf{v}_n^{selfatt}$.

We summarize the option input encodings for inference in Table ???. There are two ways to scoring with these inputs, one is to score the concatenated those vectors together, denoted as $\text{concat}(A, B) = \text{MLP}([A; B])$; The other one is scoring each of them first and then add the scores up as the final scores, such as $\text{add}(A, B) = \text{MLP}(A) + \text{MLP}(B)$.

4.3.5 Addressing Label Imbalance

From Table 4.1, we see that both client and therapist labels are imbalanced. Moreover, rarer labels are more important in both tasks. For example, it is important to identify **CT**

and **ST** utterances. For therapists, it is crucial to flag MI non-adherent (**MIN**) utterances; seasoned therapists are trained to avoid them because they correlate negatively with patient improvements. If not explicitly addressed, the frequent but less useful labels can dominate predictions.

To address this, we extend the focal loss (FL Lin et al., 2017) to the multiclass case. For a label l with probability produced by a model p_t , the loss is defined as

$$\text{FL}(p_t) = -\alpha_t(1 - p_t)^\gamma \log(p_t) \quad (4.5)$$

In addition to using a label-specific balance weight α_t , the loss also includes a modulating factor $(1 - p_t)^\gamma$ to dynamically downweight well-classified examples with $p_t \gg 0.5$. Here, the α_t 's and the γ are hyperparameters. We use FL as the default loss function for all our models.

4.4 Experiments

The original psychotherapy sessions were collected for both clinical trials and Motivational Interviewing dissemination studies including hospital settings Roy-Byrne et al. (2014), outpatient clinics Baer et al. (2009), college alcohol interventions Tollison et al. (2008); Neighbors et al. (2012); Lee et al. (2013, 2014). All sessions were annotated with the Motivational Interviewing Skills Codes (MISC) Atkins et al. (2014). We use the train/test split of Can et al. (2015); Tanana et al. (2016) to give 243 training MI sessions and 110 testing sessions. We used 24 training sessions for development. As mentioned in §??, all our experiments are based on the MISC codes grouped by Xiao et al. (2016).

4.4.1 Preprocessing and Model Setup

An MI session contains about 500 utterances on average. We use a sliding window of size $N = 8$ utterances with padding for the initial ones. We assume that we always know the identity of the speaker for all utterances. Based on this, we split the sliding windows into a client and therapist windows to train separate models. We tokenized and lower-cased utterances using spaCy Honnibal and Montani (2017). To embed words, we concatenated 300-dimensional Glove embeddings Pennington et al. (2014) with ELMo vectors Peters et al. (2018b). The appendix details the model setup and hyperparameter choices.

4.4.2 Results

Best Models. Our goal is to discover the best client and therapist models for the two tasks. We identified the following best configurations using F_1 score on the development set:

1. **Categorization:** For client, the best model does not need any word or utterance attention. For the therapist, it uses $GMGRU^H$ for word attention and $ANCHOR_{42}$ for utterance attention. We refer to these models as \mathcal{C}_C and \mathcal{C}_T respectively
2. **Forecasting:** For both client and therapist, the best model uses no word attention, and uses $SELF_{42}$ utterance attention. We refer to these models as \mathcal{F}_C and \mathcal{F}_T respectively.

Here, we show the performance of these models against various baselines. The appendix gives label-wise precision, recall and F_1 scores.

Results on Categorization. Tables 4.5 and 4.6 show the performance of the \mathcal{C}_C and \mathcal{C}_T models and the baselines. For both therapist and client categorization, we compare the best models against the same set of baselines. The majority baseline illustrates the severity of the label imbalance problem. Xiao et al. (2016), $BiGRU_{generic}$, Can et al. (2015) and Tanana et al. (2016) are the previous published baselines. The best results of previous published baselines are underlined. The last row Δ in each table lists the changes of our best model from them. $BiGRU_{ELMo}$, $CONCAT^C$, $GMGRU^H$ and $BiDAF^H$ are new baselines we define below.

The first set of baselines (above the line) do not encode dialogue history and use only the current utterance encoded with a BiGRU. The work of Xiao et al. (2016) falls in this category, and uses a 100-dimensional domain-specific embedding with weighted cross-entropy loss. Previously, it was the best model in this class. We also re-implemented this model to use either ELMo or Glove vectors with focal loss.⁵

The second set of baselines (below the line) are models that use dialogue context. Both Can et al. (2015) and Tanana et al. (2016) use well-studied linguistic features and then tagging the current utterance with both past and future utterance with CRF and MEMM, respectively. To study the usefulness of the hierarchical encoder, we implemented a model

⁵Other related work in no context exists (e.g., Pérez-Rosas et al., 2017; Gibson et al., 2017), but they either do not outperform Xiao et al. (2016) or use different data.

Method	macro	FN	CT	ST
Majority	30.6	91.7	0.0	0.0
Xiao et al. (2016)	50.0	87.9	32.8	<u>29.3</u>
BiGRU _{generic}	<u>50.2</u>	87.0	<u>35.2</u>	28.4
BiGRU _{ELMo}	52.9	87.6	39.2	32.0
Can et al. (2015)	44.0	91.0	20.0	21.0
Tanana et al. (2016)	48.3	89.0	29.0	27.0
CONCAT ^C	51.8	86.5	38.8	30.2
GMGRU ^H	52.6	89.5	37.1	31.1
BiDAF ^H	50.4	87.6	36.5	27.1
\mathcal{C}_C	53.9	89.6	39.1	33.1
$\Delta = \mathcal{C}_C - \text{score}$	+3.5	-2.1	+3.9	+3.8

Table 4.5: Main results on categorizing client codes, in terms of macro F_1 , and F_1 for each client code. Our model \mathcal{C}_C uses final dialogue vector H_n and current utterance vector v_n as input of MLP for final prediction. We found that predicting using $\text{MLP}(H_n) + \text{MLP}(v_n)$ performs better than just $\text{MLP}(H_n)$.

Method	macro	FA	RES	REC	GI	QUC	QUO	MIA	MIN
Majority	5.87	47.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Xiao et al. (2016)	59.3	<u>94.7</u>	50.2	48.3	71.9	68.7	80.1	54.0	6.5
BiGRU _{generic}	<u>60.2</u>	94.5	<u>50.5</u>	<u>49.3</u>	72.0	70.7	80.1	<u>54.0</u>	<u>10.8</u>
BiGRU _{ELMo}	62.6	94.5	51.6	49.4	70.7	72.1	80.8	57.2	24.2
Can et al. (2015)	-	94.0	49.0	45.0	<u>74.0</u>	<u>72.0</u>	<u>81.0</u>	-	-
Tanana et al. (2016)	-	94.0	48.0	39.0	69.0	68.0	77.0	-	-
CONCAT ^C	61.0	94.5	54.6	34.3	73.3	73.6	81.4	54.6	22.0
GMGRU ^H	64.9	94.9	56.0	54.4	75.5	75.7	83.0	58.2	21.8
BiDAF ^H	63.8	94.7	55.9	49.7	75.4	73.8	80.7	56.2	24.0
\mathcal{C}_T	65.4	95.0	55.7	54.9	74.2	74.8	82.6	56.6	29.7
$\Delta = \mathcal{C}_T - \text{score}$	+5.2	+0.3	+3.9	+3.8	+0.2	+2.8	+1.6	+2.6	+18.9

Table 4.6: Main results on categorizing therapist codes, in terms of macro F_1 , and F_1 for each therapist code. Models are the same as Table 4.5, but tuned for therapist codes. For the two grouped MISC set **MIA** and **MIN**, their results are not reported in the original work due to different setting.

that uses a bidirectional GRU over a long sequence of flattened utterance. We refer to this as CONCAT^C. This model is representative of the work of Huang et al. (2018), but was reimplemented to take advantage of ELMo.

For categorizing client codes, BiGRU_{ELMo} is a simple but robust baseline model. It outperforms the previous best no-context model by more than 2 points on macro F_1 . Using the dialogue history, the more sophisticated model \mathcal{C}_C further gets 1 point improvement.

Especially important is its improvement on the infrequent, yet crucial labels **CT** and **ST**. It shows a drop in the F_1 on the **FN** label, which is essentially considered to be an unimportant, background class from the point of view of assessing patient progress. For therapist codes, as the highlighted numbers in Table 4.6 show, only incorporating GMGRU-based word-level attention, GMGRU^H has already outperformed many baselines, our proposed model \mathcal{F}_T which uses both GMGRU-based word-level attention and anchor-based multi-head multihop sentence-level attention can further achieve the best overall performance. Also, note that our models outperform approaches that take advantage of future utterances.

For both client and therapist codes, concatenating dialogue history with CONCAT^C always performs worse than the hierarchical method and even the simpler $\text{BiGRU}_{\text{ELMo}}$.

Method	Dev		Test			
	CT	ST	macro	FN	CT	ST
CONCAT^F	20.4	30.2	43.6	84.4	23.0	23.5
HGRU	19.9	31.2	44.4	85.7	24.9	22.5
GMGRU^H	19.4	30.5	44.3	87.1	23.3	22.4
\mathcal{F}_C	21.1	31.3	44.3	85.2	24.7	22.7

Table 4.7: Main results on forecasting client codes, in terms of F_1 for **ST**, **CT** on dev set, and macro F_1 , and F_1 for each client code on the test set.

Method	Recall	F_1								
	R@3	macro	FA	RES	REC	GI	QUC	QUO	MIA	MIN
CONCAT^F	72.5	23.5	63.5	0.6	0.0	53.7	27.0	15.0	18.2	9.0
HGRU	76.0	28.6	71.4	12.7	24.9	58.3	28.8	5.9	17.4	9.7
GMGRU^H	76.6	26.6	72.6	10.2	20.6	58.8	27.4	6.0	8.9	7.9
\mathcal{F}_T	77.0	31.1	71.9	19.5	24.7	59.2	29.1	16.4	15.2	12.8

Table 4.8: Main results on forecasting therapist codes, in terms of Recall@3, macro F_1 , and F_1 for each label on test set

Results on Forecasting. Since the forecasting task is new, there are no published baselines to compare against. Our baseline systems essentially differ in their representation of dialogue history. The model CONCAT^F uses the same architecture as the model CONCAT^C

from the categorizing task. We also show comparisons to the simple HGRU model and the GMGRU^H model that uses a gated matchGRU for word attention.⁶

Tables ?? (a,b) show our forecasting results for client and therapist respectively. For client codes, we also report the CT and ST performance on the development set because of their importance. For the therapist codes, we also report the recall@3 to show the performance of a suggestion system that displayed three labels instead of one. The results show that even without an utterance, the dialogue history conveys signal about the next MISC label. Indeed, the performance for some labels is even better than some categorization baseline systems. Surprisingly, word attention (GMGRU^H) in Table ?? did not help in forecasting setting, and a model with the SELF₄₂ utterance attention is sufficient. For the therapist labels, if we always predicted the three most frequent labels (FA, GI, and RES), the recall@3 is only 67.7, suggesting that our models are informative if used in this suggestion-mode.

4.5 Analysis and Ablations

This section reports error analysis and an ablation study of our models on the development set. The appendix shows a comparison of pretrained domain-specific ELMo/glove with generic ones and the impact of the focal loss compared to simple or weighted cross-entropy.

4.5.1 Label Confusion and Error Breakdown

Figure 4.1 shows the confusion matrix for the client categorization task. The confusion between FN and CT/ST is largely caused by label imbalance. There are 414 CT examples that are predicted as ST and 391 examples vice versa. To further understand their confusion, we selected 100 of each for manual analysis. We found four broad categories of confusion, shown in Table 4.9.

The first category requires more complex reasoning than just surface form matching. For example, the phrase *seven out of ten* indicates that the client is very confident about changing behavior; the phrase *wind down after work* indicates, in this context, that the

⁶The forecasting task bears similarity to the next utterance selection task in dialogue state tracking work Yoshino et al. (2018). In preliminary experiments, we found that the Dual-Encoder approach used for that task consistently underperformed the other baselines described here.

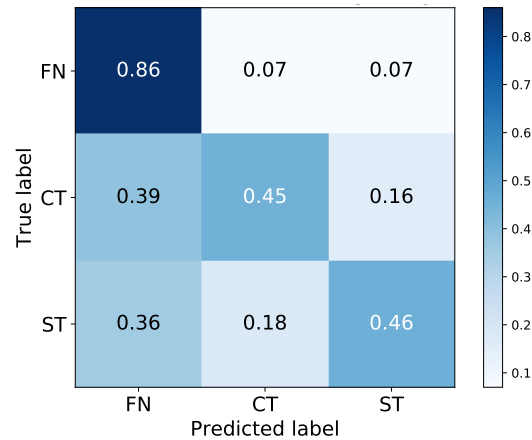


Figure 4.1: Confusion matrix for categorizing client codes, normalized by row.

Category and Explanation	Client Examples (Gold MISC)
Reasoning is required to understand whether a client wants to change behavior, even with full context (50,42)	T: On a scale of zero to ten how confident are you that you can implement this change? C: I don't know, seven maybe (CT); I have to wind down after work (ST)
Concise utterances which are easy for humans to understand, but missing information such as coreference, zero pronouns (22,31)	I mean I could try it (CT) Not a negative consequence for me (ST) I want to get every single second and minute out of it(CT)
Extremely short (≤ 5) or long sentence (≥ 40), caused by incorrect turn segmentation. (21,23)	It is a good thing (ST) Frankly, I hate it (CT) Painful (CT)
Ambivalent speech, very hard to understand even for human. (7,4)	What if it does n't work I mean what if I can't do it (ST) But I can stop whenever I want(ST)

Table 4.9: Categorization of CT/ST confusions. The two numbers in the brackets are the count of errors for predicting CT as ST and vice versa. We exemplified 100 examples for each case.

client drinks or smokes after work. We also found that the another frequent source of error is incomplete information. In a face-to-face therapy session, people may use concise and efficient verbal communication, with gestures and other body language conveying information without explaining details about, for example, coreference. With only textual context, it is difficult to infer the missing information. The third category of errors is introduced when speech is transcribed into text. The last category is about ambivalent speech. Discovering the real attitude towards behavior change behind such utterances could be difficult, even for an expert therapist.

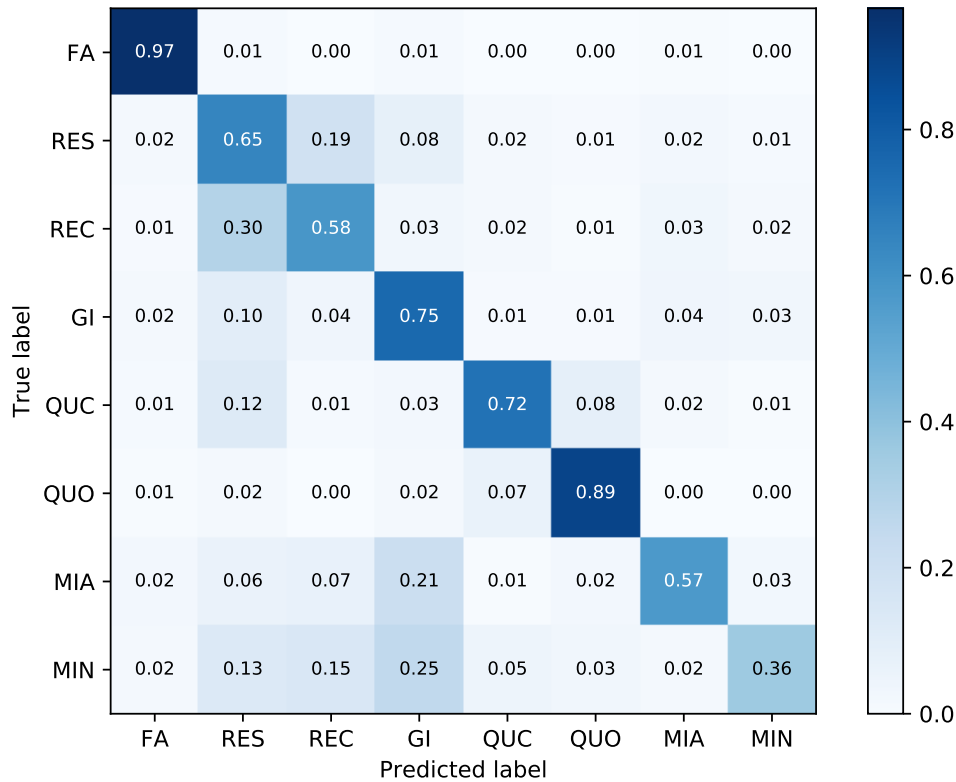


Figure 4.2: Confusion matrix for categorizing therapist codes, normalized by row.

Figures 4.1 and 4.2 show the label confusion matrices for the best categorization models. We will examine confusions that are not caused purely by a label being frequent. We observe a common confusion between the two reflection labels, **REC** and **RES**. Compared to the confusion matrix from Xiao et al. (2016), we see that our models show much-decreased confusion here. There are two reason for this confusion persisting. First, the reflections may require a much longer information horizon. We found that by increasing the window size to 16, the overall reflection results improved. Second, we need to capture richer meaning beyond surface word overlap for **RES**. We found that complex reflections usually add meaning or emphasis to previous client statements using devices such as analogies, metaphors, or similes rather than simply restating them.

Closed questions (**QUC**) and simple reflections (**RES**) are known to be a confusing set of labels. For example, an utterance like *Sounds like you're suffering?* may be both. Giving information (**GI**) is easily confused with many labels because they relate to providing information to clients, but with different attitudes. The MI adherent (**MIA**) and non-adherent (**MIN**) labels may also provide information, but with supportive or critical attitude that

Ablation	Options	macro	FN	CT	ST
history window size	0	51.6	87.6	39.2	32.0
	4	52.6	88.5	37.8	31.5
	8*	53.9	89.6	39.1	33.1
	16	52.0	89.6	39.1	33.1
word attention	+ GMGRU	52.6	89.5	37.1	31.1
	+ BiDAF	50.4	87.6	36.5	27.1
sentence attention	+ SELF ₄₂	53.9	89.2	39.1	33.2
	+ ANCHOR ₄₂	53.0	88.2	38.9	32.0

Table 4.10: Ablation study on categorizing client code. * is our best model \mathcal{C}_C . All ablation is based on it. The symbol + means adding a component to it. The default window size is 8 for our ablation models in the word attention and sentence attention parts.

may be difficult to disentangle, given the limited number of examples.

4.5.2 How Context and Attention Help?

We evaluated various ablations of our best models to see how changing various design choices changes performance. We focused on the context window size and impact of different word level and sentence level attention mechanisms. Tables 4.10 and 4.11 summarize our results.

History Size. Increasing the history window size generally helps. The biggest improvements are for categorizing therapist codes (Table 4.11), especially for the **RES** and **REC**. However, increasing the window size beyond 8 does not help to categorize client codes (Table 4.10) or forecasting (in appendix).

Word-level Attention. Only the model \mathcal{C}_T uses word-level attention. As shown in Table 4.11, when we remove the word-level attention from it, the overall performance drops by 3.4 points, while performances of **RES** and **REC** drop by 3.3 and 5 points respectively. Changing the attention to BiDAF decreases performance by about 2 points (still higher than the model without attention).

Sentence-level Attention. Removing sentence attention from the best models that have it decreases performance for the models \mathcal{C}_T and \mathcal{F}_T (in appendix). It makes little impact on the \mathcal{F}_C , however. Table 4.10 shows that neither attention helps categorizing clients codes.

Ablation	Options	macro	RES	REC	MIN
history window size	0	62.6	51.6	49.4	24.2
	4	64.4	54.3	53.2	23.7
	8*	65.4	55.7	54.9	29.7
	16	65.6	55.4	56.7	26.7
word attention	- GMGRU	62.0	51.9	51.7	16.0
	\ BiDAF	63.5	54.2	51.3	22.6
sentence attention	- ANCHOR ₄₂	64.9	56.0	54.4	21.8
	\ SELF ₄₂	63.4	55.5	48.2	21.1

Table 4.11: Ablation study on categorizing therapist codes, * is our proposed model \mathcal{C}_T . \ means substituting and – means removing that component. Here, we only report the important REC, RES labels for guiding, and the MIN label for warning a therapist.

4.5.3 Can We Suggest Empathetic Responses?

Our forecasting models are trained on regular MI sessions, according to the label distribution on Table 4.1, there are both MI adherent or non-adherent data. Hence, our models are trained to show how the therapist usually respond to a given statement.

To show whether our model can mimic *good* MI policies, we selected 35 MI sessions from our test set which were rated 5 or higher on a 7-point scale empathy or spirit. On these sessions, we still achieve a recall@3 of 76.9, suggesting that we can learn good MI policies by training on all therapy sessions. These results suggest that our models can help train new therapists who may be uncertain about how to respond to a client.

4.6 Conclusion

We addressed the question of providing real-time assistance to therapists and proposed the tasks of categorizing and forecasting MISC labels for an ongoing therapy session. By developing a modular family of neural networks for these tasks, we show that our models outperform several baselines by a large margin. Extensive analysis shows that our model can decrease the label confusion compared to previous work, especially for reflections and rare labels, but also highlights directions for future work.

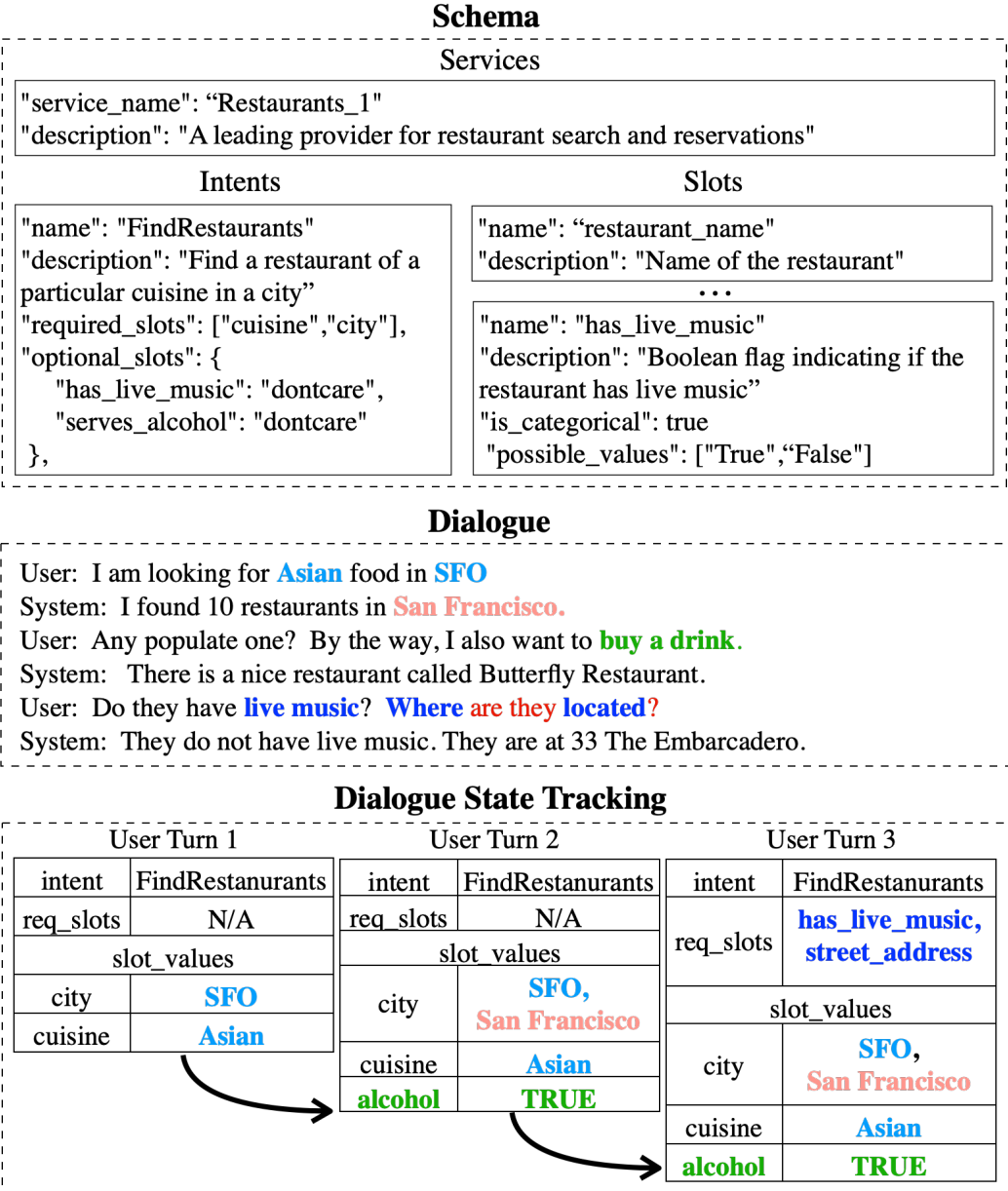
CHAPTER 5

REPRESENTING INTENT/SLOT CONCEPT WITH NATURAL LANGUAGE DESCRIPTION

From early frame-driven dialog system GUS Bobrow et al. (1977) to virtual assistants (Alexa, Siri, and Google Assistant *et al.*), frame-based dialog state tracking has long been studied to meet various challenges. In particular, how to support an ever-increasing number of services and APIs spanning multiple domains has been a focal point in recent years, evidenced by multi-domain dialog modeling Budzianowski et al. (2018); Byrne et al. (2019); Shah et al. (2018a) and transferable dialog state tracking to unseen intent/slots Mrkšić et al. (2017); Wu et al. (2019); Hosseini-Asl et al. (2020).

Recently, Rastogi et al. (2019) proposed a new paradigm called schema-guided dialog for transferable dialog state tracking by using natural language description to define a dynamic set of service schemata. As shown in Figure 5.1, the primary motivation is that these descriptions can offer effective knowledge sharing across different services, e.g., connecting semantically similar concepts across heterogeneous APIs, thus allowing a unified model to handle unseen services and APIs. With the publicly available schema-guided dialog dataset (SG-DST henceforward) as a testbed, they organized a state tracking shared task composed of four subtasks: intent classification (INTENT), requested slot identification (REQ), categorical slot labeling (CAT), and noncategorical slot labeling (NONCAT) Rastogi et al. (2020). Many participants achieved promising performance by exploiting the schema description for dialog modeling, especially on unseen services.

Despite the novel approach and promising results, current schema-guided dialog state tracking task only evaluates on a single dataset with limited variation in schema definition. It is unknown how this paradigm generalizes to other datasets and other different styles of descriptions. In this paper, we focus our investigation on the study of three aspects in schema-guided dialog state tracking: (1) schema encoding model architectures (2) sup-



- An experimental study of supplementary training on schema-guided dialog state tracking, via intermediate tasks including natural language inference and question answering.
- An in-depth analysis of different schema description styles on a new suite of benchmarking datasets with variations in schema description for both SG-DST and MULTIWOZ 2.2.

5.1 Schema-Guided Dialog State Tracking

A classic dialog state tracker predicts a dialog state frame at each user turn given the dialog history and predefined domain ontology. As shown in Figure 5.1, the key difference between schema-guided dialog state tracking and the classic paradigm is the newly added natural language descriptions. In this section, we first introduce the four subtasks and schema components in schema-guided dialog state tracking, then we outline the research questions in our paper.

Subtasks. As shown in Figure 5.1, the dialog state for each service consists of 3 parts: *active intent*, *requested slots*, *user goals* (*slot values*). Without loss of generality, for both SG-DST and MULTIWOZ 2.2 datasets, we divide their slots into categorical and non-categorical slots by following previous study on dual-strategies Zhang et al. (2019a). Thus to fill the dialog state frame for each user turn, we solve four subtasks: intent classification (INTENT), requested slot identification (REQ), categorical slot labeling (CAT), and non-categorical slot labeling (NONCAT). All subtasks require matching the current dialog history with candidate schema descriptions for multiple times.

Schema Components. Figure 5.1 shows three main schema components: service, intent, slot. For each intent, the schema also describes *optional* or *required* slots for it. For each slot, there are flags indicating whether it is categorical or not. *Categorical* means there is a set of predefined candidate values (Boolean, numeric or text). For instance, *has_live_music* in Figure 5.1 is a categorical slot with Boolean values. *Non-categorical*, on the other hand, means the slot values are filled from the string spans in the dialog history.

New Questions. These added schema descriptions pose the following three new questions. We discuss each of them in the following sections.

- Q1. How should dialogue and schema be encoded? §??
- Q2. How do different supplementary trainings impact each subtask? §??
- Q3. How do different description styles impact the state tracking performance? §5.6

5.2 Related Work

Our work is related to three lines of research: multi-sentence encoding, multi-domain and transferable dialog state tracking. However, our focus is on the comparative study of different encoder architectures, supplementary training, and schema description style variation. Thus we adopt existing strategies from multi-domain dialog state tracking.

Multi-Sentence Encoder Strategies. Similar to the recent study on encoders for response selection and article search tasks Humeau et al. (2019), we also conduct our comparative study on the two typical architectures *Cross-Encoder* Bordes et al. (2014); Lowe et al. (2015a) and *Dual-Encoder* Wu et al. (2017); Yang et al. (2018). However, they only focus on sentence-level matching tasks. All subtasks in our case require sentence-level matching between dialog context and each schema, while the non-categorical slot filling task also needs to produce a sequence of token-level representation for span detection. Hence, we study multi-sentence encoding for both sentence-level and token-level tasks. Moreover, to share the schema encoding across subtasks and turns, we also introduce a simple *Fusion-Encoder* by caching schema token embeddings in §5.4.1, which improves efficiency without sacrificing much accuracy.

Multi-domain Dialog State Tracking. Recent research on multi-domain dialog system have been largely driven by the release of large-scale multi-domain dialog datasets, such as MultiWOZ Budzianowski et al. (2018), M2M Shah et al. (2018a), accompanied by studies on key issues such as in/cross-domain carry-over Kim et al. (2019). In this paper, our goal is to understanding the design choice for schema descriptions in dialog state tracking. Thus we simply follow the in-domain cross-over strategies used in **TRADE** Wu et al. (2019). Additionally, explicit cross-domain carryover Naik et al. (2018) is difficult to generalize to new services and unknown carryover links. We use longer dialog history to inform the model on the dialog in the previous service. This simplified strategy does impact our model performance negatively in comparison to a well-designed dialog state

Datasets	Splits	Dialog	Domains	Services	Zero-shot Domains	Zero-shot Services	Function Overlapp	Collecting Method
SG-DST	Train	16142	16	26	-	-	Across/Within- domain	M2M
	Dev	2482	16	17	1	8		
	Test	4201	18	21	3	11		
MULTIWOZ 2.2	Train	9617	3	3	-	-	Across-domain	H2H
	Dev	2455	5	5	2	2		
	Test	2969	8	8	5	5		

Table 5.1: Summary of characteristics of SG-DST MULTIWOZ 2.2 datasets, in domain diversity, function overlap, data collecting methods

tracking model on seen domains. However, it helps reduce the complexity of matching extra slot descriptions for cross-service carryover. We leave the further discussion for future work.

Transferable Dialog State Tracking. Another line of research focuses on how to build a transferable dialog system that is easily scalable to newly added intents and slots. This covers diverse topics including e.g., resolving lexical/morphological variabilities by symbolic de-lexicalization-based methods Henderson et al. (2014); Williams et al. (2016), neural belief tracking Mrkšić et al. (2017), generative dialog state tracking Peng et al. (2020); Hosseini-Asl et al. (2020), modeling DST as a question answering task Zhang et al. (2019a); Lee et al. (2019); Gao et al. (2020, 2019). Our work is similar with the last class. However, we further investigate whether the DST can benefit from NLP tasks other than question answering. Furthermore, without rich description for the service/intent/slot in the schema, previous works mainly focus on simple format on question answering scenarios, such as domain-slot-type compounded names (e.g., “*restaurant-food*”), or simple question template “*What is the value for slot i?*”. We incorporate different description styles into a comparative discussion on §5.6.1.

5.3 Datasets

To the best of our knowledge, at the time of our study, SG-DST and MULTIWOZ 2.2 are the only two publicly available corpus for schema-guided dialog study. We choose both of them for our study. In this section, we first introduce these two representative datasets, then we discuss the generalizability in domain diversity, function overlapping, data collecting methods.

Schema-Guided Dialog Dataset. SG-DST dataset¹ is especially designed as a test-bed for schema-guided dialog, which contains well-designed heterogeneous APIs with overlapping functionalities between services Rastogi et al. (2019). In DSTC8 Rastogi et al. (2020), SG-DST was introduced as the standard benchmark dataset for schema-guided dialog research. SG-DST covers 20 domains, 88 intents, 365 slots.² However, previous research are mainly conducted based on this single dataset and the provided single description style. In this paper, we further extended this dataset with other benchmarking description styles as shown in §5.6, and then we perform both homogenous and heterogenous evaluation on it.

Remixed MultiWOZ 2.2 Dataset. To eliminate potential bias from the above single SG-DST dataset, we further add MULTIWOZ 2.2 Zang et al. (2020) to our study. Among various extended versions for MultiWOZ dataset (2.0-2.3, Budzianowski et al., 2018; Eric et al., 2020; Zang et al., 2020; Han et al., 2020), besides rectifying the annotation errors, MULTIWOZ 2.2 also introduced the schema-guided annotations, which covers 8 domains, 19 intents, 36 slots. To evaluate performance on seen/unseen services with MultiWOZ, we remix the MULTIWOZ 2.2 dataset to include as seen services dialogs related to *restaurant*, *attraction* and *train* during training, and eliminate slots from other domains/services from training split. For dev, we add two new domains *hotel* and *taxi* as unseen services. For test, we add all remaining domains as unseen, including those that have minimum overlap with seen services, such as *hospital*, *police*, *bus*. The statistics of data splits are shown in Appendix ?? . Note that this data split is different from the previous work on zero-shot MultiWOZ DST which takes a leave-one-out approach in Wu et al. (2019). By remixing the data in the way described above, we can evaluate the zero-shot performance on MultiWOZ in a way largely compatible with SG-DST.

Discussion. First, the two datasets cover diverse domains. MULTIWOZ 2.2 covers various possible dialogue scenarios ranging from requesting basic information about attractions through booking a hotel room or travelling between cities. While SG-DST covers more domains, such as ‘Payments’, ‘Calender’, ‘DoctorServices’ and so on.

Second, they include different levels of overlapping functionalities. SG-DST allows

¹<https://github.com/google-research-datasets/dstc8-schema-guided-dialogue>

²Please refer to the original paper for more details.

frequent function overlapping between multiple services, within the same domain (e.g. BookOneWayTicket v.s. BookRoundTripTicket), or across different domains (BusTicket v.s. TrainTicket). However, the overlapping in MULTIWOZ 2.2 only exists across different domains, e.g., ‘destination’, ‘leaveat’ slots for Taxi and Bus services, ‘pricerange’, ‘bookday’ for Restaurant and Hotel services.

Third, they are collected by two different approaches which are commonly used in dialog collecting. SG-DST is firstly collected by machine-to-machine self-play (M2M, Shah et al., 2018b) with dialog flows as seeds, then paraphrased by crowd-workers. While MULTIWOZ 2.2 are human-to-human dialogs (H2H, Kelley, 1984), which are collected with the Wizard-of-Oz approach.

We summarize the above discussion in Table 5.1. We believe that results derived from these two representative datasets can guide future research in schema guided dialog.

5.4 Dialog & Schema Representation and Inference (Q1)

In this section, we focus on the model architecture for matching dialog history with schema descriptions using pretrained BERT Devlin et al. (2019)³. To support four subtasks, we first extend *Dual-Encoder* and *Cross-Encoder* to support both sentence-level matching and token-level prediction. Then we propose an additional *Fusion-Encoder* strategy to get faster inference without sacrificing much accuracy. We summarize different architectures in Figure 5.2. Then we show the classification head and results for each subtask.

5.4.1 Encoder Architectures

Dual-Encoder. It consists of two separate BERTs to encode dialog history and schema description respectively, as Figure 5.2 (a). We follow the setting in the official baseline provided by DSTC8 Track4 Rastogi et al. (2020). We first use a fixed BERT to encode the schema description once and cached the encoded schema \mathbf{CLS}_S . Then for sentence-level representation, we concatenate dialog history representation \mathbf{CLS}_D and candidate schema representation \mathbf{CLS}_S as the whole sentence-level representation for the pair, denoted as \mathbf{CLS}^{DE} . For token-level representation, we concatenate the candidate schema

³We use BERT-base-cased for all main experiments. Other pretrained language models can be easily adapted to our study

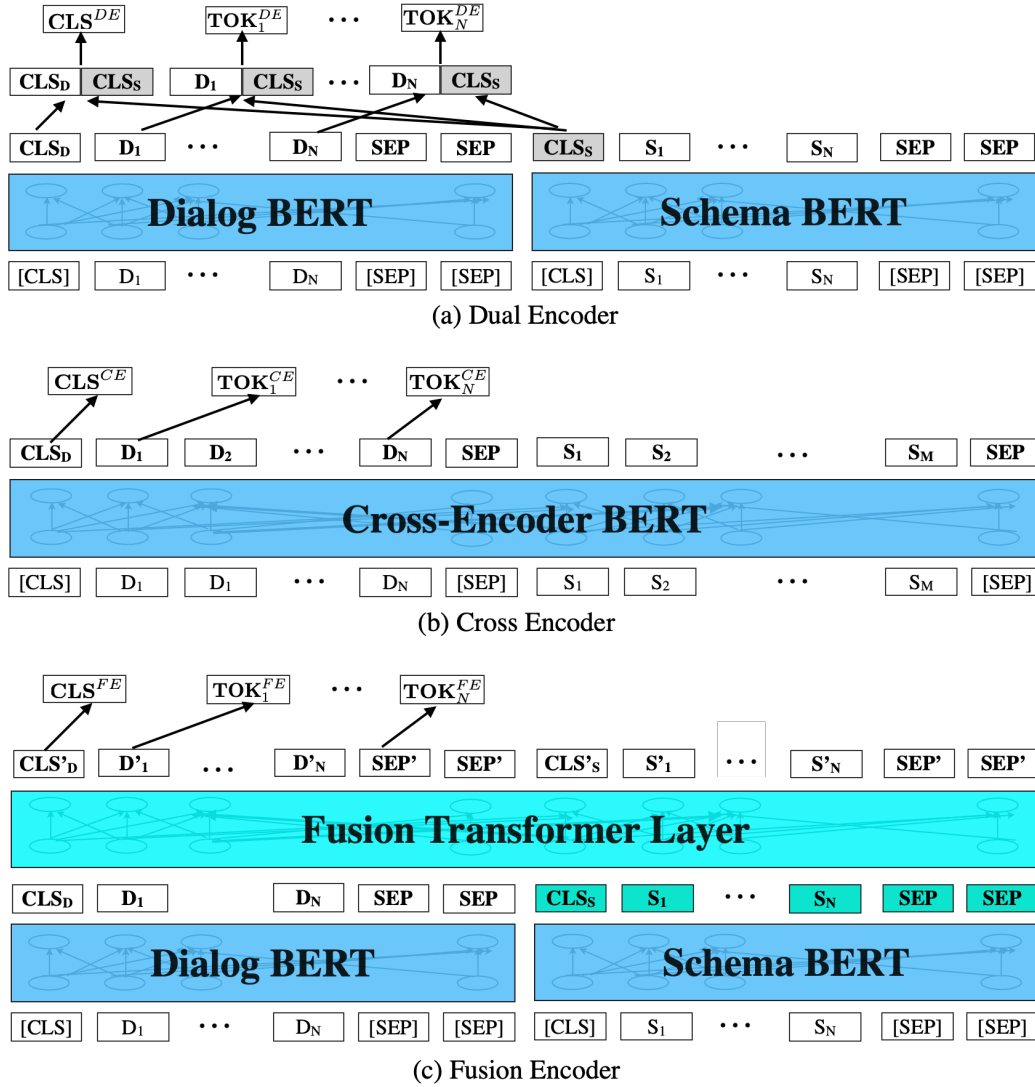


Figure 5.2: Dual-Encoder, Cross-Encoder and Fusion Encoder, shaded block will be cached during training

CLS_S with each token embedding in the dialog history, denoted as TOK^{DE} .⁴ Because the candidate schema embeddings are encoded independently from the dialog context, they can be pre-computed once and cached for fast inference.

Cross-Encoder. Another popular architecture as Figure 5.2 (b) is *Cross-Encoder*, which concatenates the dialog and schema as a single input, and encodes jointly with a single self-attentive encoder spanning over the two segments. When using BERT to encode

⁴A schema-aware dialog token embedding can also be computed by attention or other method for span-based detection tasks Humeau et al. (2019); Noroozi et al. (2020)

the concatenated sentence pair, it performs full (cross) self-attention in every transformer layers, thus offer rich interaction between the dialog and schema. BERT naturally produces a summarized representation with [CLS] embedding \mathbf{CLS}^{CE} and each schema-attended dialog token embeddings \mathbf{TOK}^{CE} . Since the dialog and schema encoding always depend on each other, it requires recomputing dialog and schema encoding for multiple times, thus much slower in inference.

Fusion-Encoder. In Figure 5.2 (c), similar to *Dual-Encoder*, *Fusion-Encoder* also encodes the schema independently with a fixed BERT and finetuning another BERT for dialog encoding. However, instead of caching a single [CLS] vector for schema representation, it caches **all token representation** for the schema including the [CLS] token. What's more, to integrate the sequences dialog token representation with schema token representation, an extra stack of transformer layers are added on top to allow token-level fusion via self-attention, similar to *Cross-Encoder*. The top transformer layers will produce embeddings for each token \mathbf{TOK}^{FE} including a schema-attended \mathbf{CLS}^{FE} of the input [CLS] from the dialog history. With cached schema token-level representations, it can efficiently produce schema-aware sentence- and token-level representation for each dialog-schema pairs.

5.4.2 Model Overview

All the above 3 encoders will produce both sentence- and token-level representations for a given sentence pair. In this section, we abstract them as two representations \mathbf{CLS} and \mathbf{TOK} , and present the universal classification heads to make decisions for each subtask.

Active Intent. To decide the intent for current dialog turn, we match current dialog history D with each intent descriptions $I_0 \dots I_k$. For each dialog-intent pair (D, I_k) , we project the final sentence-level \mathbf{CLS} representation to a single number $P_{I_k}^{active}$ with a linear layer follows a sigmoid function. We predict "NONE" if the $P_{I_k}^{active}$ of all intents are less than a threshold 0.5, which means no intent is active. Otherwise, we predict the intent with largest $P_{I_k}^{active}$. We predict the intent for each turn independently without considering the prediction on previous turns.

Requested Slot. As in Figure 5.1, multiple requested slots can exist in a single turn. We use the same strategy as in active intent prediction to predict a number P_{req}^{active} . However,

to support the multiple requested slots prediction. We predict all the requested slots with $P_{req}^{active} > 0.5$.

Categorical Slot. Categorical slots have a set of candidate values. We cannot predict unseen values via n-way classification. Instead, we do binary classification on each candidate value. Besides, rather than directly matching with values, we also need to check that whether the corresponding slot has been activated. For *Cross-Encoder* and *Fusion-Encoder*, we use typical two-stage state tracking to incrementally build the state: **Step 1.** Using **CLS** to predict the slot status as *none*, *dontcare* or *active*. When the status is *active*, we use the predicted slot value; Otherwise, it will be assigned to *dontcare* meaning no user preference for this slot, or *none* meaning no value update for the slot in current turn; **Step 2.** If Step 1 is *active*, we match the dialog history with each value and select the most related value by ranking. We train on cross entropy loss. Two-stage strategy is efficient for *Dual-Encoder* and *Fusion-Encoder*, where cached schema can be reused, and get efficiently ranked globally in a single batch. However, it is not scalable for *Cross-Encoder*, especially for large number of candidate values in MultiWOZ dataset. Hence, during training, we only use a binary cross-entropy for each single value and postpone the ranking only to the inference time.

Noncategorical Slot. The slot status prediction for noncategorical slot use the same two-stage strategy. Besides that, we use the token representation of dialog history **TOK** to compute two softmax scores f_{start}^i and f_{end}^i for each token i , to represent the score of predicting the token as start and end position respectively. Finally, we find the valid span with maximum sum of the start and end scores.

5.4.3 Experiments on Encoder Comparison

To fairly compare all three models, we follow the same schema input setting as in Table 5.2. We trained separate models for SG-DST and the remixed MultiWOZ datasets for all the experiments in our papers⁵. Because there are very few intent and requested slots in MultiWOZ 2.2 dataset, we ignore the intent and requested slots tasks for MultiWOZ 2.2 in our paper.

Results. As shown in Table 5.3, *Cross-Encoder* performs the best over all subtasks. Our

⁵Appendix ?? shows the detailed experiment setup

Intent	service description, intent description
Req	service description, slot description
Cat	slot description, cat value
NonCat	service description, slot description

Table 5.2: Schema description input used for different tasks to compare *Dual-Encoder*, *Cross-Encoder*, and *Fusion-Encoder*. In the appendix ??, we also studies other compositions of description input. We found that service description will not help for INTENT, REQ and CAT tasks, while the impact on NONCAT task also varies from SG-DST and MULTIWOZ 2.2 dataset.

Method/Task	SG-DST					MULTIWOZ 2.2		
	Acc	F1	Joint Acc			Joint Acc		
	Intent	Req	Cat	NonCat	All	Cat	NonCat	All
Seen Services								
Dual-Encoder	94.51	99.62	87.92	47.77	43.20	79.20	79.34	65.64
Fusion-Encoder	94.90	99.69	88.94	48.78	58.52	81.37	80.58	67.43
Cross-Encoder	95.55	99.59	93.68	91.85	87.58	85.99	81.02	71.93
Unseen Services								
Dual-Encoder	89.73	95.20	42.44	31.62	19.51	56.92	50.82	31.83
Fusion-Encoder	90.47	95.95	48.79	35.91	22.85	57.01	52.23	33.64
Cross-Encoder	93.84	98.26	71.55	74.13	54.54	59.85	59.62	38.46

Table 5.3: Test set results on SG-DST and MULTIWOZ 2.2. The *Dual-Encoder* model is a re-implementation of official DSTC8 baseline from Rastogi et al. (2019). Other models are trained with the architecture described in our paper.

Fusion-Encoder with partial attention outperforms the *Dual-Encoder* by a large margin, especially on categorical and noncategorical slots predictions. Additionally, on seen services, we found that *Dual-Encoder* and *Fusion-Encoder* can perform as good as *Cross-Encoder* on INTENT and REQ tasks. However, they cannot generalize well on unseen services as *Cross-Encoder*.

Inference Speed. To test the inference speed, we conduct all the experiments with a maximum affordable batch size to fully exploit 2 V100 GPUs (with 16GB GPU RAM each). During training, we log the inference time of each evaluation on dev set. Both *Dual-Encoder* and *Fusion-Encoder* can do joint inference across 4 subtasks to obtain an integral dialog state for a dialog turn example. *Dual-Encoder* achieves the highest inference speed of **603.35** examples per GPU second, because the encoding for dialog and schema are fully separated. A dialog only needed to be encoded for once during the inference of a dialog

	SG-DST											
	intent			req			cat			noncat		
	all	seen	unseen	all	seen	unseen	all	seen	unseen	all	seen	unseen
Δ_{SNLI}	+0.51	+0.02	+0.68	-0.19	+0.38	-0.38	-1.63	-2.87	-1.23	-4.7	-0.1	-6.25
Δ_{SQuAD}	-1.81	-0.17	-1.32	-0.25	-0.01	-0.33	-2.87	-3.02	-5.17	+1.99	-1.79	+3.25

Table 5.4: Relative performance improvement of different supplementary training on SG-DST dataset

	MULTIWOZ 2.2						
	cat			noncat			
	all	seen	unseen	all	seen	unseen	
Δ_{SNLI}	+2.05	+0.6	-0.7	+3.64	+1.05	+4.84	
Δ_{SQuAD}	+0.04	-0.71	+0.41	+1.93	-2.21	+4.27	

Table 5.5: Relative performance improvement of different supplementary training on MULTIWOZ 2.2 dataset

state example while the schema are precomputed once. However, for *Cross-Encoder*, to predict a dialog state for a single turn, it need to encode more than 300 sentence pairs in a batch, thus only processes **4.75** examples per GPU second. *Fusion-Encoder* performs one time encoding on dialog history, but it needs to jointly encode the same amount of dialog-schema pairs as *Cross-Encoder*, instead, however, with a two-layer transformer encoder. Overall it achieves **10.54** examples per GPU second, which is **2.2x** faster than *Cross-Encoder*. With regarding to the accuracy in Table 5.3, *Fusion-Encoder* performs much better than *Dual-Encoder*, especially on unseen services.

5.5 Supplementary Training (Q2)

Besides the pretrain-fintune framework used in §??, Phang et al. (2018) propose to add a supplementary training phase on an intermediate task after the pretraining, but before finetuning on target task. It shows significant improvement on the target tasks. Moreover, large amount pretrained and finetuned transformer-based models are publicly accessible, and well-organized in model hubs for sharing, training and testing⁶. Given the new task of schema-guided dialog state tracking, in this section, we study our four subtasks with

⁶e.g., Huggingface(<https://huggingface.co/models>) and ParlAL(<https://parl.ai/docs/zoo.html>), etc.

different intermediate tasks for supplementary training.

5.5.1 Intermediate Tasks

As described in § 5.4.2, all our 4 subtasks take a pair of dialog and schema description as input, and predict with the summerized sentence-pair **CLS** representation. While NON-CAT also requires span-based detection such as question answering. Hence, they share the similar problem structure with the following sentence-pair encoding tasks.

Natural Language Inference. Given a hypothesis/premise sentence pair, natural language inference is a task to determine whether a hypothesis is entailed, contradicted or neutral given that premise. **Question Answering.** Given a passage/question pairs, the task is to extract the span-based answer in the passage.

Hence, when finetuning BERT on our subtasks, instead of directly using the originally pretrained BERT, we use the BERT finetuned on the above two tasks for further finetuning. Due to better performance of *Cross-Encoder* in §??, we directly use the finetuned *Cross-Encoder* version of BERT models on SNLI and SQuAD2.0 dataset from Huggingface model hub. We add extra speaker tokens [user:] and [system:] into the vocabulary for encoding the multi-turn dialog histories.

5.5.2 Results on Supplementary Training

Table ?? shows the performances gain when finetuning 4 subtasks based on models with the above SNLI and SQuAD2.0 supplementary training.

We mainly find that SNLI helps on INTENT task, SQuAD2 mainly helps on NON-CAT task, while neither of them helps much on CAT task. Recently, Namazifar et al. (2020) also found that when modeling dialog understanding as question answering task, it can benefit from a supplementary training on SQuAD2 dataset, especially on few-shot scenarios, which is a similar findings as our NONCAT task. Result difference on REQ task is minor, because it is a relatively easy task, adding any supplementary training did n't help much. Moreover, for CAT task, the sequence 2 of the input pair is the slot description with a categorical slot value, thus the meaning overlapping between the full dialog history and the slot/value is much smaller than SNLI tasks. On the other side, **CLS** token in SQuAD BERT is finetuned for null predictions via start and end token classifiers, which is different

from the the single CLS classifier in CAT task.

5.6 Impact of Description Styles (Q3)

Previous work on schema-guided dialog Rastogi et al. (2020) are only based on the provided descriptions in SG-DST dataset. Recent work on modeling dialog state tracking as reading comprehension Gao et al. (2019) only formulate the descriptions as simple question format with existing intent/slot names, it is unknown how it performs when compared to other description styles. Moreover, they only conduct homogeneous evaluation where training and test data share the same description style. In this section, We also investigate how a model trained on one description style will perform on other different styles, especially in a scenario where chat-bot developers may design their own descriptions. We first introduce different styles of descriptions in our study, and then we train models on each description style and evaluate on tests with corresponding homogeneous and heterogeneous styles of descriptions. Given the best performance of *Cross-Encoder* shown in the previous section and its popularity in DSTC8 challenges, we adopt it as our model architecture in this section.

5.6.1 Benchmarking Styles

For each intent/slot, we describe their functionalities by the following different descriptions styles:

IDENTIFIER. This is the least informative case of name-based description: we only use meaningless intent/slot identifiers, e.g. Intent_1, Slot_2. It means we don't use description from any schema component. We want to investigate how a simple identifier-based description performs in schema-guided dialog modeling, and the performance lower-bound on transferring to unseen services.

NAMEONLY. Using the original intent/slot names in SG-DST and MULTIWOZ 2.2 dataset as descriptions, to show whether name is enough for schema-guided dialog modeling.

Q-NAME. This is corresponding to previous work by Gao et al. (2019). For each intent/slot, it generate a question to inquiry about the intent and slot value of the dialog. For each slot, it simply follows the template '*What is the value for slot i?*'. Besides that, our work also extend the intent description by following the template "*Is the user intending to intent j*".

ORIG. The original descriptions in SG-DST and MULTIWOZ 2.2 dataset.

Q-ORIG. Different from the **Q-NAME**, firstly it is based on the original descriptions; secondly, rather than always use the “what is” template to inquiry the intent/slot value, We add “what”, “which”, “how many” or “when” depending on the entity type required for the slot. Same as **Q-NAME**, we just add prefixes as “Is the user intending to...” in front of the original description. In a sum, this description is just adding question format to original description. The motivation of this description is to see whether the question format is helpful or not for schema-guided dialog modeling.

To test the model robustness, we also create two paraphrased versions **NAME-PARA** and **ORIG-PARA** for **NAMEONLY** and **ORIG** respectively. We first use nematus Sennrich et al. (2017) to automatically paraphrase the description with back translation, from English to Chinese and then translate back, then we manually check the paraphrase to retain the main meaning. Appendix ?? shows examples for different styles of schema descriptions.

5.6.2 Results on Description Styles

Unlike the composition used in Table 5.2, we don’t use the service description to avoid its impact. For each style, we train separate models on 4 subtasks, then we evaluate them on different target styles. First, Table 5.6 summarizes the performance for homogeneous evaluation, while Table 5.7 shows how the question style description can benefit from SQuAD2 finetuning. Then we also conduct heterogeneous evaluation on the other styles⁷ as shown in Table 5.8.

5.6.2.1 Homogeneous Evaluation

Is name-based description enough? As shown in Table 5.6, **IDENTIFIER** is the worst case of using name description, its extremely bad performance indicates name-based description can be very unstable. However, we found that simple meaningful name-based description actually can perform the best in INTENT and REQ task, and they perform worse on CAT and NONCAT tasks comparing to the bottom two *rich* descriptions.⁸ After careful

⁷We don’t consider the meaningless **IDENTIFIER** style due to its bad performance

⁸Only exception happens in CAT on MULTIWOZ 2.2. When creating MULTIWOZ 2.2 Zang et al. (2020), the slots with less than 50 different slot values are classified as categorical slots, which leads to inconsistencies.

Style\Task	SG-DST				MULTIWOZ 2.2	
	Intent	Req	Cat	NonCat	Cat	NonCat
IDENTIFIER	61.16	91.48	62.47	30.19	34.25	52.28
NAMEONLY	94.24	98.84	74.01	75.63	53.72	56.18
Q-NAME	93.31	98.86	74.36	74.86	54.19	56.17
ORIG	93.01	98.55	74.51	75.76	52.19	57.20
Q-ORIG	93.42	98.51	76.64	76.60	53.61	57.80

Table 5.6: Homogeneous evaluation results of different description style on SG-DST dataset and MULTIWOZ 2.2 datasets. The middle horizontal line separate the two name-based descriptions and two rich descriptions in our settings. All numbers in the table are mixed performance including both seen and unseen services.

analysis on the intents in SG-DST datasets, we found that most services only contains two kinds of intents, an information retrieval intent with a name prefix "Find-", "Get-", "Search-"; another transaction intent like "Add-", "Reserve-" or "Buy-". Interestingly, we found that all the intent names in the original schema-guided dataset strictly follows an action-object template with a composition of words without abbreviation, such as "FindEvents", "BuyEventTickets". This simple name template is good enough to describe the core functionality of an intent in SG-DST dataset.⁹ Additionally, REQ is a relatively simpler task, requesting information are related to special attributes, such as "has_live_music", "has_wifi", where keywords co-occured in the slot name and in the user utterance, hence rich explanation cannot help further. On the other side, *rich* descriptions are more necessary for CAT and NONCAT task. Because in many cases, slot names are too simple to represent the functionalities behind it, for example, slot name "passengers" cannot fully represent the meaning "number of passengers in the ticket booking".

Does question format help? As shown in Table 5.6, when comparing row **Q-ORIG** v.s. **ORIG**, we found extra question format can improve the performance on CAT and NONCAT task on both SG-DST and MULTIWOZ 2.2 datasets, but not for INTENT and REQ tasks. We believe that question format helps the model to focus more on specific entities in the dialog history. However, when adding a simple question pattern to **NAMEONLY**, comparing row

We put detailed discuss about MULTIWOZ 2.2 in the supplementary material

⁹This action-object template has also been found efficient for open domain intent induction task(e.g., Vedula et al., 2020, OPINE).

Style/Dataset	SG-DST			MULTIWOZ 2.2		
	all	seen	unseen	all	seen	unseen
ORIG	+1.99	-1.79	+3.25	+1.93	-2.21	+4.27
Q-ORIG	+6.13	-2.01	+8.84	+1.06	-1.28	+3.06
NAMEONLY	-0.45	-1.49	-0.11	+1.75	+0.58	+1.77
Q-NAME	+0.05	-2.98	+1.04	-0.04	-0.32	+1.25

Table 5.7: Performance changes when using BERT finetuned on SQuAD2 dataset to further finetuning on our NONCAT task.

Q-NAME and **NAMEONLY**, there is no consistent improvement on both of the two datasets. Further more, we are curious about whether BERT finetuned on SQuAD2 (SQuAD2-BERT) can further help on the question format. Because NONCAT are similar with span-based question answering, we focus on NONCAT here. Table 5.7 shows that, after applying the supplementary training on SQuAD2 (§??), almost all models get improved on unseen splits however slightly dropped on seen services. Moreover, comparing to **Q-NAME**, **Q-ORIG** is more similar to the natural questions in the SQuAD2, we observe that **Q-ORIG** gains more than **Q-NAME** from pretrained model on SQuAD2.

5.6.2.2 Heterogeneous

In this subsection, we first simulate a scenario when there is no recommended description style for the future unseen services. Hence, unseen services can follow any description style in our case. We average the evaluation performance on three other descriptions and summarized in Table 5.8. The Δ column shows the performance change compared to the homogeneous performance. It is not surprising that almost all models perform worse on heterogeneous styles than on homogeneous styles due to different distribution between training and evaluation. The bold number shows the best average performance on heterogeneous evaluation for each subtask. The trends are similar with the analysis in homogeneous evaluation 5.6.2.1, the name-based descriptions perform better than other rich descriptions on intent classification tasks. While on other tasks, the **ORIG** description performs more robust, especially on NONCAT task.

Furthermore, we consider another scenario where fixed description convention such as **NAMEONLY** and **ORIG** are suggested to developers, they must obey the basic style convention but still can freely use their own words, such as abbreviation, synonyms,

adding extra modifiers. We train each model on **NAMEONLY** and **ORIG**, then evaluate on the corresponding paraphrased version respectively. In the last two rows of Table 5.8, the column ‘para’ shows performance on paraphrased schema, while Δ shows the performance change compared to the homogeneous evaluation. **ORIG** still performs more robust than **NAMEONLY** when schema descriptions get paraphrased on unseen services.

Style\Task	SG-DST							
	Intent(Acc)		Req(F1)		Cat(Joint Acc)		NonCat(Joint Acc)	
	mean	Δ	mean	Δ	mean	Δ	mean	Δ
NAMEONLY	82.47	-11.47	96.92	-1.64	61.37	-5.54	56.53	-14.68
Q-NAME	93.27	+0.58	97.88	-0.76	68.55	+2.63	62.92	-6.30
ORIG	79.47	-12.70	97.42	-0.74	68.58	-0.3	66.72	-3.11
Q-ORIG	84.57	-8.24	96.70	-1.45	68.40	-2.89	56.17	-15.00
	para	Δ	para	Δ	para	Δ	para	Δ
NAMEONLY	92.22	-1.74	97.69	-0.87	67.39	-0.7	67.17	-4.04
ORIG	91.54	-0.63	98.42	+0.26	71.74	+2.86	67.68	-2.16

Table 5.8: Results on unseen service with heterogeneous description styles on SG-DST dataset. More results and qualitative analysis are in the appendix ??

5.7 Conclusion

In this paper, we studied three questions on schema-guided dialog state tracking: encoder architectures, impact of supplementary training, and effective schema description styles. The main findings are as follows:

By caching the token embedding instead of the single CLS embedding, a simple partial-attention *Fusion-Encoder* can achieve much better performance than *Dual-Encoder*, while still infers two times faster than *Cross-Encoder*. We quantified the gain via supplementary training on two intermediate tasks. By carefully choosing representative description styles according to recent works, we are the first of doing both homogeneous/heterogeneous evaluations for different description style in schema-guided dialog. The results show that simple name-based description performs well on INTENT and REQ tasks, while NON-CAT tasks benefits from richer styles of descriptions. All tasks suffer from inconsistencies in description style between training and test, though to varying degrees.

Our study are mainly conducted on two datasets: SG-DST and MULTIWOZ 2.2, while the speed-accuracy balance of encoder architectures and the findings in supplementary training are expected to be dataset-agnostic, because they depend more on the nature of the subtasks than the datasets. Based on our proposed benchmarking descriptions suite, the

homogeneous and heterogeneous evaluation has shed the light on the robustness of cross-style schema-guided dialog modeling, we believe our study will provide useful insights for future research.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

6.1 Claims and Research Contribution Revisited

6.2 Future Work Direction

[*Structured Prediction Problem Beyond Text*_{TODO}]

Computational Biology. igenomes, 3D structure. protein-protein interactions.

Computer Vision,

Multimodality

6.3 Summary

APPENDIX A

THE FIRST

This is an appendix. Notice that the L^AT_EX markup for an appendix is, surprisingly, `\chapter`. The `\appendix` command does not produce a heading; instead, it just changes the numbering style from numeric to alphabetic, and it changes the heading prefix from **CHAPTER** to **APPENDIX**.

Blah blah blah blah blah blah blah blah blah blah blah blah blah.
Blah blah blah blah blah blah blah blah blah blah blah blah blah. Blah
blah blah blah blah blah blah blah blah blah blah blah blah. Blah blah
blah blah blah blah blah blah blah blah blah blah blah blah. Blah blah
blah blah blah blah blah blah blah blah blah blah blah blah. Blah blah blah
blah blah blah blah blah blah blah blah blah blah blah blah. Blah blah
blah blah blah blah blah blah blah blah blah blah blah blah. Blah blah
blah blah blah blah blah blah blah blah blah blah blah blah. Blah blah
blah blah blah blah blah blah blah blah blah blah blah blah. Blah blah
blah blah blah blah blah blah blah blah blah blah blah.

APPENDIX B

THE SECOND

This is an appendix.

Blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah.
Blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah. Blah
blah blah blah blah blah blah blah blah blah blah blah blah blah blah. Blah blah
blah blah blah blah blah blah blah blah blah blah blah blah blah. Blah blah
blah blah blah blah blah blah blah blah blah blah blah blah. Blah blah blah
blah blah blah blah blah blah blah blah blah blah blah. Blah blah blah
blah blah blah blah blah blah blah blah blah blah. Blah blah blah blah
blah blah blah blah blah blah blah blah blah. Blah blah blah blah blah
blah blah blah blah blah blah blah blah. Blah blah blah blah blah blah
blah blah blah blah blah blah.

APPENDIX C

THE THIRD

This is an appendix.

REFERENCES

- Omri Abend and Ari Rappoport. 2013a. Universal Conceptual Cognitive Annotation (UCCA). In *Proceedings of the 51th Meeting of the Association for Computational Linguistics*, pages 228–238, Sofia, Bulgaria.
- Omri Abend and Ari Rappoport. 2013b. Universal conceptual cognitive annotation (ucca). In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 228–238.
- Omri Abend and Ari Rappoport. 2017. The state of the art in semantic representation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 77–89.
- Yoav Artzi, Kenton Lee, and Luke Zettlemoyer. 2015. Broad-coverage CCG Semantic Parsing with AMR. *EMNLP*.
- David C Atkins, Mark Steyvers, Zac E Imel, and Padhraic Smyth. 2014. Scaling up the evaluation of psychotherapy: evaluating motivational interviewing fidelity via statistical text classification. *Implementation Science*, 9(1):49.
- John S Baer, Elizabeth A Wells, David B Rosengren, Bryan Hartzler, Blair Beadnell, and Chris Dunn. 2009. Agency context and tailored training in technology transfer: A pilot evaluation of motivational interviewing training for community counselors. *Journal of substance abuse treatment*, 37(2):191–202.
- Collin F Baker, Charles J Fillmore, and John B Lowe. 1998. The berkeley framenet project. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, pages 86–90.
- Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013a. Abstract Meaning Representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria.
- Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013b. Abstract Meaning Representation for Sembanking. *LAW@ACL*.
- Jonathan Baxter. 2000. A model of inductive bias learning. *Journal of artificial intelligence research*, 12:149–198.
- Kenneth R Beesley and Lauri Karttunen. 2003. Finite-state morphology: Xerox tools and techniques. *CSLI, Stanford*.
- David Belanger and Andrew McCallum. 2016. Structured prediction energy networks. In *International Conference on Machine Learning*, pages 983–992. PMLR.

- Elie Bienenstock, Stuart Geman, and Daniel Potter. 1996. Compositionality, mdl priors, and object recognition. In *NIPS*.
- Daniel G Bobrow, Ronald M Kaplan, Martin Kay, Donald A Norman, Henry Thompson, and Terry Winograd. 1977. Gus, a frame-driven dialog system. *Artificial intelligence*, 8(2):155–173.
- Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.
- Antoine Bordes, Jason Weston, and Nicolas Usunier. 2014. Open question answering with weakly supervised embedding models. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 165–180. Springer.
- Peter F Brown, Vincent J Della Pietra, Stephen A Della Pietra, and Robert L Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, 19(2):263–311.
- Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic. 2018. Multiwoz-a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026.
- Brian L Burke, Christopher W Dunn, David C Atkins, and Jerry S Phelps. 2004. The emerging evidence base for motivational interviewing: A meta-analytic and qualitative inquiry. *Journal of Cognitive Psychotherapy*, 18(4):309–322.
- Bill Byrne, Karthik Krishnamoorthi, Chinnadhurai Sankar, Arvind Neelakantan, Daniel Duckworth, Semih Yavuz, Ben Goodrich, Amit Dubey, Andy Cedilnik, and Kyu-Young Kim. 2019. Taskmaster-1: Toward a realistic and diverse dialog dataset. *arXiv preprint arXiv:1909.05358*.
- Doğan Can, David C Atkins, and Shrikanth S Narayanan. 2015. A dialog act tagging approach to behavioral coding: A case study of addiction counseling conversations. In *Sixteenth Annual Conference of the International Speech Communication Association*.
- Jie Cao, Michael Tanana, Zac Imel, Eric Poitras, David Atkins, and Vivek Srikumar. 2019a. Observing dialogue in therapy: Categorizing and forecasting behavioral codes. In *Proceedings of ACL 2019*.
- Jie Cao, Yi Zhang, Adel Youssef, and Vivek Srikumar. 2019b. Amazon at mrp 2019: Parsing meaning representations with lexical and phrasal anchoring. In *Proceedings of the Shared Task on MRP at the CoNLL 2019*.
- Lynn Carlson, Daniel Marcu, and Mary Ellen Okurowski. 2003. Building a discourse-tagged corpus in the framework of rhetorical structure theory. In *Current and new directions in discourse and dialogue*, pages 85–112. Springer.
- Wanxiang Che, Longxu Dou, Yang Xu, Yuxuan Wang, Yijia Liu, and Ting Liu. 2019. HIT-SCIR at MRP 2019: A unified pipeline for meaning representation parsing via efficient

- training and effective encoding. In *Proceedings of the Shared Task on Cross-Framework Meaning Representation Parsing at the 2019 Conference on Natural Language Learning*, pages 76–85, Hong Kong, China.
- Wei-Te Chen and Martha Palmer. 2017. Unsupervised amr-dependency parse alignment. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, volume 1, pages 558–567.
- Kenneth Mark Colby. 1975. *Artificial Paranoia: A Computer Simulation of Paranoid Process*. Pergamon Press.
- Michael Collins. 2003. Head-driven statistical models for natural language parsing. *Computational linguistics*, 29(4):589–637.
- Franck Dernoncourt and Ji Young Lee. 2017. Pubmed 200k rct: a dataset for sequential sentence classification in medical abstracts. *IJCNLP 2017*, page 308.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.
- Lucia Donatelli, Meaghan Fowlie, Jonas Groschwitz, Alexander Koller, Matthias Lindemann, Mario Mina, and Pia Weißenhorn. 2019. Saarland at MRP 2019: Compositional parsing across all graphbanks. In *Proceedings of the Shared Task on Cross-Framework Meaning Representation Parsing at the 2019 Conference on Natural Language Learning*, pages 66–75, Hong Kong, China.
- Timothy Dozat and Christopher D Manning. 2016. Deep biaffine attention for neural dependency parsing. *ICLR*.
- Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyang Gao, Adarsh Kumar, Anuj Goyal, Peter Ku, and Dilek Hakkani-Tur. 2020. Multiwoz 2.1: A consolidated multi-domain dialogue dataset with state corrections and state tracking baselines. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 422–428.
- Jeffrey Flanigan, Sam Thomson, Jaime G Carbonell, Chris Dyer, and Noah A Smith. 2014. A Discriminative Graph-Based Parser for the Abstract Meaning Representation. *ACL*.
- Andrea Galassi, Marco Lippi, and Paolo Torrioni. 2019. Attention, please! a critical review of neural attention models in natural language processing. *arXiv preprint arXiv:1902.02181*.
- Shuyang Gao, Sanchit Agarwal, Tagyoung Chung, Di Jin, and Dilek Hakkani-Tur. 2020. From machine reading comprehension to dialogue state tracking: Bridging the gap. *arXiv preprint arXiv:2004.05827*.

- Shuyang Gao, Abhishek Sethi, Sanchit Agarwal, Tagyoung Chung, Dilek Hakkani-Tur, and Amazon Alexa AI. 2019. Dialog state tracking: A neural reading comprehension approach. In *20th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, page 264.
- Samuel J. Gershman. 2021. *What Makes Us Smart*. Princeton University Press.
- James Gibson, Dogan Can, Panayiotis Georgiou, David C Atkins, and Shrikanth S Narayanan. 2017. Attention networks for modeling behaviors in addiction counseling. In *Proceedings of the 2016 Conference of the International Speech Communication Association INTERSPEECH*.
- Sonal Gupta, Rushin Shah, Mrinal Mohit, Anuj Kumar, and Mike Lewis. 2018. Semantic parsing for task oriented dialog using hierarchical representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2787–2792, Brussels, Belgium. Association for Computational Linguistics.
- Jan Hajic, Eva Hajicová, Jarmila Panevová, Petr Sgall, Ondrej Bojar, Silvie Cinková, Eva Fucíková, Marie Mikulová, Petr Pajas, Jan Popelka, et al. 2012. Announcing prague czech-english dependency treebank 2.0. In *LREC*, pages 3153–3160.
- Ting Han, Ximing Liu, Ryuichi Takanobu, Yixin Lian, Chongxuan Huang, Wei Peng, and Minlie Huang. 2020. Multiwoz 2.3: A multi-domain task-oriented dataset enhanced with annotation corrections and co-reference annotation. *arXiv e-prints*, pages arXiv–2010.
- Harry F Harlow. 1949. The formation of learning sets. *Psychological review*, 56(1):51.
- Matthew Henderson, Blaise Thomson, and Steve Young. 2014. Word-based dialog state tracking with recurrent neural networks. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 292–299.
- Daniel Hershcovich, Zohar Aizenbud, Leshem Choshen, Elinor Sulem, Ari Rappoport, and Omri Abend. 2019. SemEval-2019 task 1: Cross-lingual semantic parsing with UCCA. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 1–10, Minneapolis, Minnesota, USA. Association for Computational Linguistics.
- Daniel Hershcovich and Ofir Arviv. 2019. TUPA at MRP 2019: A multi-task baseline system. In *Proceedings of the Shared Task on Cross-Framework Meaning Representation Parsing at the 2019 Conference on Natural Language Learning*, pages 28–39, Hong Kong, China.
- Matthew Honnibal and Ines Montani. 2017. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear*.
- Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. 2020. A simple language model for task-oriented dialogue. *arXiv preprint arXiv:2005.00796*.
- Xiaolei Huang, Lixing Liu, Kate Carey, Joshua Woolley, Stefan Scherer, and Brian Borsari. 2018. Modeling temporality of human intentions by domain adaptation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 696–701.

- Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2019. Poly-encoders: Architectures and pre-training strategies for fast and accurate multi-sentence scoring. In *ICLR*.
- Zac E Imel, Derek D Caperton, Michael Tanana, and David C Atkins. 2017. Technology-enhanced human interaction in psychotherapy. *Journal of counseling psychology*, 64(4):385.
- Angelina Ivanova, Stephan Oepen, Lilja Øvrelid, and Dan Flickinger. 2012. Who did what to whom?: A contrastive study of syntacto-semantic dependencies. In *Proceedings of the sixth linguistic annotation workshop*, pages 2–11. Association for Computational Linguistics.
- Wei Jiang, Yu Zhang, Zhenghua Li, and Min Zhang. 2019. Hlt@suda at semeval 2019 task 1: UCCA graph parsing as constituent tree parsing. *arXiv preprint arXiv:1903.04153*.
- Dan Jurafsky and James H Martin. 2019. *Speech and language processing*. Pearson.
- Pavan Kapanipathi, Ibrahim Abdelaziz, Srinivas Ravishankar, Salim Roukos, Alexander Gray, Ramón Fernández Astudillo, Maria Chang, Cristina Cornelio, Saswati Dana, Achille Fokoue-Nkoutche, et al. 2021. Leveraging abstract meaning representation for knowledge base question answering. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3884–3894.
- John F Kelley. 1984. An iterative design methodology for user-friendly natural language office information applications. *ACM Transactions on Information Systems (TOIS)*, 2(1):26–41.
- Sungdong Kim, Sohee Yang, Gyuwan Kim, and Sang-Woo Lee. 2019. Efficient dialogue state tracking by selectively overwriting memory. *arXiv preprint arXiv:1911.03906*.
- Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations*.
- Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *ICLR*.
- Nikita Kitaev and Dan Klein. 2018. Constituency parsing with a self-attentive encoder. *arXiv preprint arXiv:1805.01052*.
- Ioannis Konstas, Srinivasan Iyer, Mark Yatskar, Yejin Choi, and Luke Zettlemoyer. 2017. Neural amr: Sequence-to-sequence models for parsing and generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 146–157.
- Brenden M. Lake, Tomer David Ullman, Joshua B. Tenenbaum, and Samuel J. Gershman. 2016. Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40.
- Christine M Lee, Jason R Kilmer, Clayton Neighbors, David C Atkins, Cheng Zheng, Denise D Walker, and Mary E Larimer. 2013. Indicated prevention for college student marijuana use: A randomized controlled trial. *Journal of consulting and clinical psychology*, 81(4):702.

- Christine M Lee, Clayton Neighbors, Melissa A Lewis, Debra Kaysen, Angela Mittmann, Irene M Geisner, David C Atkins, Cheng Zheng, Lisa A Garberson, Jason R Kilmer, et al. 2014. Randomized controlled trial of a spring break intervention to reduce high-risk drinking. *Journal of consulting and clinical psychology*, 82(2):189.
- Hwaran Lee, Jinsik Lee, and Tae-Yoon Kim. 2019. Sumbt: Slot-utterance matching for universal and scalable belief tracking. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5478–5483.
- Jiwei Li, Minh-Thang Luong, and Dan Jurafsky. 2015. A hierarchical neural autoencoder for paragraphs and documents. *arXiv preprint arXiv:1506.01057*.
- Tao Li and Vivek Srikumar. 2019. Augmenting Neural Networks with First-order Logic. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.
- Zuchao Li, Hai Zhao, Zhuosheng Zhang, Rui Wang, Masao Utiyama, and Eiichiro Sumita. 2019. SJTU–NICT at MRP 2019: Multi-task learning for end-to-end uniform semantic graph parsing. In *Proceedings of the Shared Task on Cross-Framework Meaning Representation Parsing at the 2019 Conference on Natural Language Learning*, pages 45–54, Hong Kong, China.
- Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988.
- Fei Liu, Jeffrey Flanigan, Sam Thomson, Norman Sadeh, and Noah A Smith. 2015. Toward abstractive summarization using semantic representations. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1077–1086.
- Ryan Lowe, Nissan Pow, Iulian Serban, and Joelle Pineau. 2015a. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In *The 16th Annual SIGdial Meeting on Discourse and Dialogue*.
- Ryan Lowe, Nissan Pow, Iulian V. Serban, and Joelle Pineau. 2015b. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In *Proceedings of SIGDIAL*.
- Brad W Lundahl, Chelsea Kunz, Cynthia Brownell, Derrik Tollefson, and Brian L Burke. 2010. A meta-analysis of motivational interviewing: Twenty-five years of empirical studies. *Research on social work practice*, 20(2):137–160.
- Minh-Thang Luong, Ilya Sutskever, Quoc V Le, Oriol Vinyals, and Wojciech Zaremba. 2014. Addressing the rare word problem in neural machine translation. *IJCNLP*.
- Chunchuan Lyu and Ivan Titov. 2018. Amr parsing as graph prediction with latent alignment. *ACL*.
- Renata K Martins and Daniel W McNeil. 2009. Review of motivational interviewing in promoting health behaviors. *Clinical psychology review*, 29(4):283–293.

- J May. 2016. SemEval-2016 Task 8: Meaning Representation Parsing. In *Proceedings of SemEval*, pages 1063–1073, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Gonzalo Mena, David Belanger, Scott Linderman, and Jasper Snoek. 2018. Learning latent permutations with gumbel-sinkhorn networks. *arXiv preprint arXiv:1802.08665*.
- George A Miller. 1998. *WordNet: An electronic lexical database*. MIT press.
- William Miller and Stephen Rollnick. 2003. Motivational interviewing: Preparing people for change. *Journal for Healthcare Quality*, 25(3):46.
- William R Miller, Theresa B Moyers, Denise Ernst, and Paul Amrhein. 2003. Manual for the motivational interviewing skill code (misc). *Unpublished manuscript. Albuquerque: Center on Alcoholism, Substance Abuse and Addictions, University of New Mexico*.
- William R Miller and Stephen Rollnick. 2012. *Motivational interviewing: Helping people change*. Guilford press.
- Tom M Mitchell. 1980. *The need for biases in learning generalizations*. Department of Computer Science, Laboratory for Computer Science Research.
- Yusuke Miyao, Stephan Oepen, and Daniel Zeman. 2014. In-house: An ensemble of pre-existing off-the-shelf parsers. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 335–340.
- Nikola Mrkšić, Diarmuid Ó Séaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve Young. 2017. Neural belief tracker: Data-driven dialogue state tracking. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1777–1788.
- Chetan Naik, Arpit Gupta, Hancheng Ge, Mathias Lambert, and Ruhi Sarikaya. 2018. Contextual slot carryover for disparate schemas. *Proc. Interspeech 2018*, pages 596–600.
- Mahdi Namazifar, Alexandros Papangelis, Gokhan Tur, and Dilek Hakkani-Tür. 2020. Language model is all you need: Natural language understanding as question answering. *arXiv preprint arXiv:2011.03023*.
- Clayton Neighbors, Christine M Lee, David C Atkins, Melissa A Lewis, Debra Kaysen, Angela Mittmann, Nicole Fossos, Irene M Geisner, Cheng Zheng, and Mary E Larimer. 2012. A randomized controlled trial of event-specific prevention strategies for reducing problematic drinking associated with 21st birthday celebrations. *Journal of consulting and clinical psychology*, 80(5):850.
- Vahid Noroozi, Yang Zhang, Evelina Bakhturina, and Tomasz Kornuta. 2020. A fast and robust bert-based dialogue state tracker for schema-guided dialogue dataset. *Workshop on Conversational Systems Towards Mainstream Adoption at KDD 2020*.
- Stephan Oepen, Omri Abend, Jan Hajič, Daniel Hershcovich, Marco Kuhlmann, Tim O’Gorman, Nianwen Xue, Jayeol Chun, Milan Straka, and Zdeňka Urešová. 2019. MRP 2019: Cross-framework Meaning Representation Parsing. In *Proceedings of the Shared Task on Cross-Framework Meaning Representation Parsing at the 2019 Conference on Natural Language Learning*, pages 1–27, Hong Kong, China.

- Stephan Oepen and Dan Flickinger. 2019. The ERG at MRP 2019: Radically compositional semantic dependencies. In *Proceedings of the Shared Task on Cross-Framework Meaning Representation Parsing at the 2019 Conference on Natural Language Learning*, pages 40–44, Hong Kong, China.
- Stephan Oepen, Marco Kuhlmann, Yusuke Miyao, Daniel Zeman, Silvie Cinková, Dan Flickinger, Jan Hajič, and Zdeňka Urešová. 2015. SemEval 2015 Task 18. Broad-coverage semantic dependency parsing. In *Proceedings of the 9th International Workshop on Semantic Evaluation*, page 915–926, Bolder, CO, USA.
- Stephan Oepen, Marco Kuhlmann, Yusuke Miyao, Daniel Zeman, Dan Flickinger, Jan Hajič, Angelina Ivanova, and Yi Zhang. 2014. SemEval 2014 Task 8. Broad-coverage semantic dependency parsing. In *Proceedings of the 8th International Workshop on Semantic Evaluation*, page 63–72, Dublin, Ireland.
- Stephan Oepen and Jan Tore Lønning. 2006. Discriminant-based mrs banking. In *LREC*, pages 1250–1255.
- Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational linguistics*, 31(1):71–106.
- George Papandreou and Alan Yuille. 2011. Perturb-and-map random fields: Using discrete optimization to learn and sample from energy models–iccv 2011 paper supplementary material–. *ICCV*.
- Baolin Peng, Chunyuan Li, Jinchao Li, Shahin Shayandeh, Lars Liden, and Jianfeng Gao. 2020. Soloist: Few-shot task-oriented dialog with a single pre-trained auto-regressive model. *arXiv*, pages arXiv–2005.
- Xiaochang Peng, Linfeng Song, and Daniel Gildea. 2015. A Synchronous Hyperedge Replacement Grammar based approach for AMR parsing. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 32–41, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.
- Verónica Pérez-Rosas, Rada Mihalcea, Kenneth Resnicow, Satinder Singh, Lawrence Ann, Kathy J Goggin, and Delwyn Catley. 2017. Predicting counselor behaviors in motivational interviewing encounters. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, volume 1, pages 1128–1137.
- Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018a. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.
- Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018b. Deep contextualized word representations. In *Proc. of NAACL*.

- Jason Phang, Thibault F  vry, and Samuel R Bowman. 2018. Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks. *arXiv preprint arXiv:1811.01088*.
- Nima Pourdamghani, Yang Gao, Ulf Hermjakob, and Kevin Knight. 2014. Aligning English Strings with Abstract Meaning Representation Graphs. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 425–429, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind K Joshi, and Bonnie L Webber. 2008. The penn discourse treebank 2.0. In *LREC*. Citeseer.
- Michael Pust, Ulf Hermjakob, Kevin Knight, Daniel Marcu, and Jonathan May. 2015. Using Syntax-Based Machine Translation to Parse English into Abstract Meaning Representation. *arXiv.org*.
- Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2019. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. *AAAI 2019*.
- Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. Schema-guided dialogue state tracking task at dstc8. *Workshop on DSTC8, AAAI 2020*.
- Bob Rehder. 2003. A causal-model theory of conceptual representation and categorization. *Journal of experimental psychology. Learning, memory, and cognition*, 29 6:1141–59.
- Peter Roy-Byrne, Kristin Bumgardner, Antoinette Krupski, Chris Dunn, Richard Ries, Dennis Donovan, Imara I West, Charles Maynard, David C Atkins, Meredith C Graves, et al. 2014. Brief intervention for problem drug use in safety-net primary care settings: a randomized clinical trial. *Jama*, 312(5):492–501.
- Jost Schatzmann, Kallirroi Georgila, and Steve Young. 2005. Quantitative evaluation of user simulation techniques for spoken dialogue systems. In *6th SIGdial Workshop on DISCOURSE and DIALOGUE*.
- Craig S Schwalbe, Hans Y Oh, and Allen Zweben. 2014. Sustaining motivational interviewing: a meta-analysis of training studies. *Addiction (Abingdon, England)*, 109(8):1287–94.
- Rico Sennrich, Orhan Firat, Kyunghyun Cho, Alexandra Birch, Barry Haddow, Julian Hitschler, Marcin Junczys-Dowmunt, Samuel L  ubli, Antonio Valerio Miceli Barone, Jozef Mokry, and Maria N  dejde. 2017. Nematus: a toolkit for neural machine translation. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 65–68, Valencia, Spain. Association for Computational Linguistics.
- Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. In *ICLR*.
- Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *AAAI*, volume 16, pages 3776–3784.

- Pararth Shah, Dilek Hakkani-Tür, Bing Liu, and Gokhan Tür. 2018a. Bootstrapping a neural conversational agent with dialogue self-play, crowdsourcing and on-line reinforcement learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*, pages 41–51, New Orleans - Louisiana. Association for Computational Linguistics.
- Pararth Shah, Dilek Hakkani-Tür, Gokhan Tür, Abhinav Rastogi, Ankur Bapna, Neha Nayak, and Larry Heck. 2018b. Building a conversational agent overnight with dialogue self-play. *arXiv preprint arXiv:1801.04871*.
- Linfeng Song, Daniel Gildea, Yue Zhang, Zhiguo Wang, and Jinsong Su. 2019. Semantic neural machine translation using amr. *Transactions of the Association for Computational Linguistics*, 7:19–31.
- Alessandro Sordoni, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob Grue Simonsen, and Jian-Yun Nie. 2015. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 553–562. ACM.
- Elizabeth S. Spelke. 1990. Principles of object perception. *Cognitive Science*, 14:29–56.
- Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational linguistics*, 26(3):339–373.
- Ida Szubert, Adam Lopez, and Nathan Schneider. 2018. A structured syntax-semantics interface for english-amr alignment. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1169–1180.
- Michael Tanana, Kevin A Hallgren, Zac E Imel, David C Atkins, and Vivek Srikumar. 2016. A comparison of natural language processing methods for automated coding of motivational interviewing. *Journal of substance abuse treatment*, 65:43–50.
- Sean J Tollison, Christine M Lee, Clayton Neighbors, Teryl A Neil, Nichole D Olson, and Mary E Larimer. 2008. Questions and reflections: the use of motivational interviewing microskills in a peer-led brief alcohol intervention for college students. *Behavior Therapy*, 39(2):183–194.
- Lifu Tu and Kevin Gimpel. 2018. Learning approximate inference networks for structured prediction. *ICLR*.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. 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.
- Nikhita Vedula, Nedim Lipka, Pranav Maneriker, and Srinivasan Parthasarathy. 2020. Open intent extraction from natural language interactions. In *Proceedings of The Web Conference 2020*, pages 2009–2020.

- Chuan Wang and Nianwen Xue. 2017. Getting the most out of amr parsing. In *Proceedings of the 2017 conference on empirical methods in natural language processing*, pages 1257–1268.
- Chuan Wang, Nianwen Xue, and Sameer Pradhan. 2015. Boosting Transition-based AMR Parsing with Refined Actions and Auxiliary Analyzers. *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 1–6.
- Wenhui Wang, Nan Yang, Furu Wei, Baobao Chang, and Ming Zhou. 2017. Gated self-matching networks for reading comprehension and question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 189–198.
- Joseph Weizenbaum. 1966. ELIZA – a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1):36–45.
- Jason Williams, Antoine Raux, and Matthew Henderson. 2016. The dialog state tracking challenge series: A review. *Dialogue & Discourse*, 7(3):4–33.
- Florian Wolf and Edward Gibson. 2005. Representing discourse coherence: A corpus-based study. *Computational linguistics*, 31(2):249–287.
- David H Wolpert, William G Macready, et al. 1995. No free lunch theorems for search. Technical report, Technical Report SFI-TR-95-02-010, Santa Fe Institute.
- Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. 2019. Transferable multi-domain state generator for task-oriented dialogue systems. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 808–819.
- Yu Wu, Wei Wu, Chen Xing, Ming Zhou, and Zhoujun Li. 2017. Sequential matching network: A new architecture for multi-turn response selection in retrieval-based chatbots. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 496–505.
- Bo Xiao, Dogan Can, James Gibson, Zac E Imel, David C Atkins, Panayiotis G Georgiou, and Shrikanth S Narayanan. 2016. Behavioral coding of therapist language in addiction counseling using recurrent neural networks. In *Proceedings of the 2016 Conference of the International Speech Communication Association INTERSPEECH*, pages 908–912.
- Liu Yang, Minghui Qiu, Chen Qu, Jiafeng Guo, Yongfeng Zhang, W Bruce Croft, Jun Huang, and Haiqing Chen. 2018. Response ranking with deep matching networks and external knowledge in information-seeking conversation systems. *arXiv preprint arXiv:1805.00188*.
- Koichiro Yoshino, Chiori Hori, Julien Perez, Luis Fernando D’Haro, Lazaros Polymenakos, Chulaka Gunasekara, Walter S. Lasecki, Jonathan Kummerfeld, Michael Galley, Chris Brockett, Jianfeng Gao, Bill Dolan, Sean Gao, Tim K. Marks, Devi Parikh, and Dhruv Batra. 2018. The 7th dialog system technology challenge. *arXiv preprint*.
- Xiaoxue Zang, Abhinav Rastogi, Srinivas Sunkara, Raghav Gupta, Jianguo Zhang, and Jindong Chen. 2020. MultiWOZ 2.2 : A dialogue dataset with additional annotation

corrections and state tracking baselines. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 109–117, Online. Association for Computational Linguistics.

Jian-Guo Zhang, Kazuma Hashimoto, Chien-Sheng Wu, Yao Wan, Philip S Yu, Richard Socher, and Caiming Xiong. 2019a. Find or classify? dual strategy for slot-value predictions on multi-domain dialog state tracking. *arXiv*, pages arXiv–1910.

Justine Zhang, Jonathan P Chang, Cristian Danescu-Niculescu-Mizil, Lucas Dixon, Yiqing Hua, Nithum Thain, and Dario Taraborelli. 2018. Conversations Gone Awry: Detecting Early Signs of Conversational Failure. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*.

Sheng Zhang, Xutai Ma, Kevin Duh, and Benjamin Van Durme. 2019b. AMR Parsing as Sequence-to-Graph Transduction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Florence, Italy. Association for Computational Linguistics.

BINOMIAL NOMENCLATURE INDEX

A

Alces alces 1, 3–6
Antilocapra americana 1, 3–6

B

bobcat (*Lynx rufus*) 4, 5
 bongo (*Tragelaphus eurycerus*) 4, 5
Bos taurus indicus 1, 5

C

Cervus canadensis 1, 4, 5
Connochaetes gnou 1, 9–11
 cougar (*Puma concolor*) 4, 5

E

E. coli bacterium 1, 9–11
 elk (*Cervus canadensis*) 4, 5

J

jaguar (*Panthera onca*) 4, 5

L

Lama glama 1, 4, 5
 llama (*Lama glama*) 4, 5
 lynx (*Lynx canadensis*) 4, 5
Lynx canadensis 4, 5
Lynx rufus 5

M

manatee (*Trichechus inunguis*) 3–6
 margay (*Leopardus wiedii*) 3–6
 marmot (*Marmota marmota*) 3–6
Marmota marmota 3–6
Marmota monax 1, 5
 moose (*Alces alces*) 3–6

O

ocelot (*Leopardus pardalis*) 3–6
Odocoileus virginianus 1, 5

P

Panthera onca 4, 5

Panthera tigris 1, 5
 pronghorn (*Antilocapra americana*) 3–6
Puma concolor 5

T

tiger (*Panthera tigris*) 1, 5
Tragelaphus eurycerus 4, 5

V

Vicugna vicugna 1, 5
 vicuña (*Vicugna vicugna*) 1, 5

W

wapiti (*Cervus canadensis*) 1
 white-tailed deer (*Odocoileus virginianus*) 1, 5
 woodchuck (*Marmota monax*) 1, 5

Z

zebu (cattle) (*Bos taurus indicus*) 1, 5

TOPIC INDEX

- | | |
|--|---|
| <p>gnu hair, 14, 76–78
 gnu hair , 14, 76–78</p> <p>aardvark, 37
 aardwolf, 37
 addax, 37
 African ungulate, <i>see</i> gnu
 <i>Alces alces</i>, <i>see</i> moose, 14, 18, 37, 55
 antelope, 37
 <i>Antilocapra americana</i>, <i>see</i> pronghorn, 14, 18, 37, 55
 aoudad, 37
 azymous (unleavened), 37</p> <p>beaver, 37
 bison, 37
 bobcat (<i>Lynx rufus</i>), 37
 bongo (<i>Tragelaphus eurycerus</i>), 37
 Borel measure (μ), 90
 <i>Bos taurus indicus</i>, 14</p> <p>caribou, 37
 <i>Cervus canadensis</i>, <i>see</i> elk, <i>see</i> wapiti, 14, 37
 cheetah, 37
 <i>Connochaetes gnou</i>, <i>see</i> gnu, 14, 76–78
 cougar (<i>Puma concolor</i>), 37
 coyote, 37
 crocodile, 37</p> <p>DCT, <i>see</i> discrete cosine transform
 deer, 37
 DWT, <i>see</i> discrete wavelet transform</p> <p><i>E. coli</i> bacterium, 14, 76–78
 elephant, 37
 elk (<i>Cervus canadensis</i>), 37
 <i>Escherichia coli</i>, <i>see</i> <i>E. coli</i></p> <p>ferret, 37
 fox, 37
 free software, <i>see also</i> GNU Project</p> <p>gazelle, 37
 gecko, 37</p> | <p>gila monster, 37
 giraffe, 37
 gnu, 14, 18, 37, 55, 76, 77, 78
 diet, 14, 18, 37, 55, 77, 78
 dry season, 14, 18, 37, 55, 77, 78
 wet season, 14, 18, 37, 55, 77, 78
 hair, 14, 76, 78
 color, 14, 76, 78
 DNA analysis, 14, 76, 78
 texture, 14, 76, 78
 thickness, 14, 76, 78
 predators, 14, 37, 77, 78
 crocodiles, 14, 37, 77, 78
 hyenas, 14, 37, 77, 78
 lions, 14, 37, 77, 78
 gnu hair, 14, 76–78
 gnu hair, 14, 76–78
 GNU Project, 14, 76, 77, 78</p> <p>hartebeest, 37
 hippopotamus, 37
 hyena, 37</p> <p>ibex, 37
 impala, 37</p> <p>jackal, 37
 jackass, 37
 jackrabbit, 37
 jaguar (<i>Panthera onca</i>), 37
 jerboa, 37</p> <p>kangaroo, 37
 koala, 37
 kudu, 37</p> <p><i>Lama glama</i>, <i>see</i> llama, 14, 37
 lemming, 37
 lemur, 37
 leopard, 37
 <i>Leopardus pardalis</i>, <i>see</i> ocelot
 <i>Leopardus wiedii</i>, <i>see</i> margay
 lion, 37</p> |
|--|---|

- llama (*Lama glama*), 37
 lynx (*Lynx canadensis*), 37
Lynx canadensis, see lynx, 37
Lynx rufus, see bobcat
- mammoth, 18, 37, 55
 manatee (*Trichechus inunguis*), 18, 37, 55
 margay (*Leopardus wiedii*), 18, 37, 55
 marmot (*Marmota marmota*), 18, 37, 55
Marmota marmota, 18, 37, 55
Marmota monax, see woodchuck, 14
 mastiff, 18, 37, 55
 mastodon, 18, 37, 55
 moose (*Alces alces*), 18, 37, 55
 μ (mu), see Borel measure
 musk ox, 18, 37, 55
 muskrat, 18, 37, 55
- narwhal, 18, 37, 55
 nautilus, 18, 37, 55
 Neanderthal, 18, 37, 55
 nilgai, 18, 37, 55
- ocelot (*Leopardus pardalis*), 18, 37, 55
 octopus, 18, 37, 55
Odocoileus virginianus, 14
 okapi, 18, 37, 55
 opossum, 18, 37, 55
- pachyderm, see elephant, see hippopotamus,
 see rhinoceros
 panda, 18, 37, 55
 panther, 18, 37, 55
Panthera onca, see jaguar, 37
Panthera tigris, see tiger, 14
 peccary, 18, 37, 55
 pronghorn (*Antilocapra americana*), 18, 37, 55
 pterodactyl, 18, 37, 55
 puffin, 18, 37, 55
 puma, see cougar
Puma concolor, see cougar
- quagga, 18, 37, 55
 quail, 18, 37, 55
 Queensland viper, 18, 37, 55
- raccoon, 18, 37, 55
 rat, 18, 37, 55
 rhea, 18, 37, 55
 Rhine berry, 18, 37, 55
- Rhinegrave, 18, 37, 55
 Rhineland, 18, 37, 55
 rhinestone, 18, 37, 55
 rhino, see rhinoceros
 rhinocerite, 18, 37, 55
 rhinoceros, 18, 37, 55
 rhinoceros horn, 18, 37, 55
 rhinodon, 18, 37, 55
 rhinodont, 18, 37, 55
- saber-toothed cat, 14
 salamander, 14
 sapajou, 14
 skink, 14
 skunk, 14
 sunfish, 14
 suslik, 14
 swordfish, 14
- tangun, 14
 tapir, 14
 tiger (*Panthera tigris*), 14
 toucan, 14
Tragelaphus eurycerus, see bongo, 37
 transform, see also Discrete DCT Transform,
 see also Fast Fourier Transform
Trichechus inunguis, see manatee
 tuna, 14
 turbot, 14
- ungulate, see gnu, see impala, see kudu, see
 springbok
 unicorn, 14
- vampire bat, 14
Vicugna vicugna, see vicuña, 14
 vicuña (*Vicugna vicugna*), 14
 vulture, 14
- wallaby, 14
 walrus, 14
 wanderoo, 14
 wapiti (*Cervus canadensis*), 14
 wart hog, 14
 water buffalo, 14
 weasel, 14
 whale, 14
 whippet, 14
 white-tailed deer (*Odocoileus virginianus*), 14

wildebeest, *see* gnu

wolf, 14

wolverine, 14

woodchuck (*Marmota monax*), 14

X radiation, 14

X ray, 14

xenon (noble gas), 14

xylem (woody tissue of a plant), 14

xylophone, 14

yak, 14

yucca, 14

zebra, 14

zebu (cattle) (*Bos taurus indicus*), 14

zoetrope, 14

zythum (ancient Egyptian malt beverage), 14