Multi-domain Spoken Language Understanding Using Domain- and Task-aware Parameterization

LIBO QIN, FUXUAN WEI, and MINHENG NI, Harbin Institute of Technology, China
YUE ZHANG, Westlake University, China
WANXIANG CHE, YANGMING LI, and TING LIU, Harbin Institute of Technology, China

Spoken language understanding (SLU) has been addressed as a supervised learning problem, where a set of training data is available for each domain. However, annotating data for a new domain can be both financially costly and non-scalable. One existing approach solves the problem by conducting multi-domain learning where parameters are shared for joint training across domains, which is *domain-agnostic* and *task-agnostic*. In the article, we propose to improve the parameterization of this method by using domain-specific and task-specific model parameters for fine-grained knowledge representation and transfer. Experiments on five domains show that our model is more effective for multi-domain SLU and obtain the best results. In addition, we show its transferability when adapting to a new domain with little data, outperforming the prior best model by 12.4%. Finally, we explore the strong pre-trained model in our framework and find that the contributions from our framework do not fully overlap with contextualized word representations (RoBERTa).

CCS Concepts: • **Computing methodologies → Natural language processing;**

Additional Key Words and Phrases: Multi-domain spoken language understanding, domain-specific and task-specific model, fine-grained knowledge representation and transfer

ACM Reference format:
Libo Qin, Fuxuan Wei, Minheng Ni, Yue Zhang, Wanxiang Che, Yangming Li, and Ting Liu. 2022. Multi-domain Spoken Language Understanding Using Domain- and Task-aware Parameterization. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* 21, 4, Article 77 (January 2022), 17 pages.
https://doi.org/10.1145/3502198

1 INTRODUCTION

Spoken language understanding (SLU) [48] plays an important role in task-oriented dialog systems. It consists of two typical subtasks, including intent detection and slot filling [40]. For example, in Figure 1, given an input utterance "*I want to watch action movie*," the outputs consist of an

This work was supported by Westlake-BrightDreams Robotics research grant. Besides, this work was also supported by the National Key R&D Program of China via grant 2020AAA0106501 and the National Natural Science Foundation of China (NSFC) via grants 61976072 and 61772153. This work was also supported by the Zhejiang Lab's International Talent Fund for Young Professionals.
Authors' addresses: L. Qin, F. Wei, M. Ni, W. Che, Y. Li, and T. Liu, Harbin Institute of Technology, Research Center for Social Computing and Information Retrieval, Harbin, Heilongjiang, 150001, China; emails: {lbqin, fuxuanwei, minhengni, car, yangmingli, tliu}@ir.hit.edu.cn; Y. Zhang (corresponding author), Westlake University, Westlake Institute for Advanced Study, Hangzhou, Zhejiang, 310024, China; email: yue.zhang@wias.org.cn.

2375-4699/2022/01-ART77 $15.00
https://doi.org/10.1145/3502198

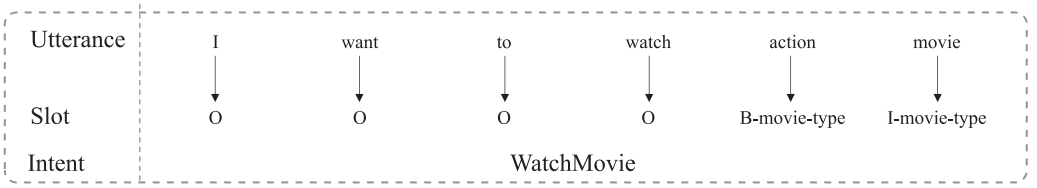


Fig. 1. An example with intent and slot annotation (BIO format).

overall intent class label (i.e., `WatchMovie`) and a slot label sequence (i.e., `O`, `O`, `O`, `O`, `B-movie-type`, `I-movie-type`). In particular, the former is a classification task, and the latter can be addressed using sequence labeling. Since slots highly depend on the intent information, dominant SLU systems in the literature [3, 4, 17, 21, 27, 43] adopt joint models for the two tasks, we follow this line of work by jointly solving intent detection and slot filling.

Intuitively, there exists a wide range of business domains (e.g., *watch movie*, *book ticket*) with shared and specific characteristics, and it can be infeasible to train a model for each domain. In practice, a dialogue model should handle multiple domains. To this end, some existing work has endeavored towards using resources from all domains to train a model [9, 13]. As shown in Figure 2(a), Kim et al. [13] build a model by combining labeled data from different domains for jointly training intent detection and slot filling. Their models use the same set of model parameters for representing both cross-domain and cross-task information. While this can be useful for feature integration, the method is *domain-agnostic* and *task-agnostic*: (1) *domain-agnostic*: One set of shared parameters cannot effectively distinguish the domain-shared and domain-specific features, which limits their performance. (2) *task-agnostic*: The method represents knowledge for the sentence-level intent detection task and token-level slot-filling task equally by using a unified network, and therefore does not offer fine-grained channels for learning task-specific knowledge. Take the sub-sentence "*watch action movie*" for example, the shared and domain-specific knowledge on the words "*watch*" and "*action movie*" is subtle, because "*action movie*" is domain-specific while "*watch*" can be shared with other domains [45]. Therefore, domain-specific tokens should be represented with more knowledge from a specific domain, while domain shared tokens should keep shared characteristics across domains. Unfortunately, solely relying on a unified framework cannot achieve the fine-trained domain knowledge representation and transfer, which greatly limits its transferability when a new domain with little data is given.

To this end, we propose a domain-aware and task-aware model (i.e., multi-level shared-private framework) for multi-domain joint intent detection and slot filling, which is shown in Figure 2(b). To solve the *domain-agnostic* issue, we first propose to use a standard shared-private framework [20] as a foundation, which consists of a domain-shared module for representing common knowledge across domains and a domain-specific module for explicitly extracting specific features for each domain. In addition, utterances from different domains have different sentence syntactic patterns, which helps the model to capture domain-aware features. Thus, we explore the domain-aware syntax information and we empirically find modeling syntax information can substantially improve multi-domain SLU.

To address the *task-agnostic* issue, we extend the vanilla shared-private framework to a multi-level structure, achieving the fine-grained domain knowledge transfer: (1) sentence-level domain knowledge transfer is achieved by using a *sentence-level* shared-private architecture for modeling *intents*; (2) token-level domain knowledge transfer is achieved by using a *token-level* shared-private mechanism for modeling *slots*. Besides, a *slot filter* is applied to each token to selectively decide which tokens receive private representation in addition to a domain-shared representation. A *slot*

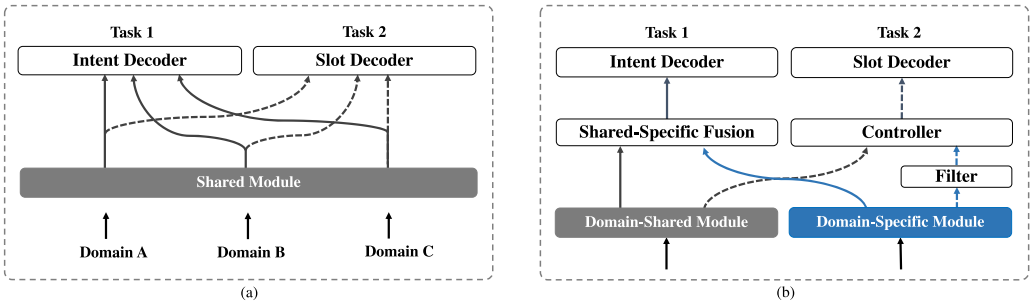


Fig. 2. Methods for multi-domain spoken language understanding. (a) Prior work trains a single model on a mixed dataset. (b) Our proposed domain-aware and task-aware model. Dashed line denotes information flow to slot filling, and solid line denotes information flow to intent detection. Blue color represents Domain-specific Module, and gray color denotes Domain-shared Module. Better viewed in color.

controller is further introduced to control the weights between domain-shared and private token representations, achieving the fine-grained combination of domain knowledge.

We conduct experiments on two benchmarks, MTOD [35] and ASMixed [1, 4], including five different domains in total. Experiments show that our method achieves state-of-the-art results, with a 91.27% sentence accuracy on the MTOD dataset, outperforming the prior best result by 1.81%. On the ASMixed dataset, we achieve 84.81% sentence accuracy, outperforming the prior best result by 3.33%. Besides, given a new domain with little labeled data, our framework can effectively transfer knowledge from source training domains, thereby outperforming the existing best model by 12.4%.

Finally, Pre-trained models (PLMs) have achieved surprising results across almost all NLP tasks. A natural question is raised whether our framework can still obtain improvement over the pre-trained model. To answer this question, we explore the pre-trained model (RoBERTa) [22] in our framework and find that our framework works orthogonally with pre-trained model.

In summary, the contributions of our work can be concluded as follows:

- We propose a domain-specific and task-specific parameterize method (i.e., multi-level shared-private framework) for multi-domain SLU, the structure of which is domain-aware and task-aware, which greatly improves the transferability across domains.
- We propose a *token-level* shared-private mechanism, which enables the model to achieve a fine-grained domain knowledge fusion for slot filling. To the best of our knowledge, this is the first attempt to consider the fine-grained knowledge transfer for multi-domain SLU.
- Experiments on two benchmarks show that our framework obtains substantial improvement over existing multi-domain SLU methods and achieves state-of-the-art performance.
- We explore and analyze the effect of incorporating pre-trained model (RoBERTa) in multi-domain SLU tasks and empirically show that the contributions from our framework do not fully overlap with contextualized word representations.

2 TASK DEFINITION

Intent Detection: Given input utterance $X = (x_1, \ldots, x_n)$ (n denotes the length of X), **intent detection (ID)** can be considered as a sentence classification task to decide the intent label o^I, which is formulated as:

$$o^I = \texttt{Intent-Detection}(X). \tag{1}$$

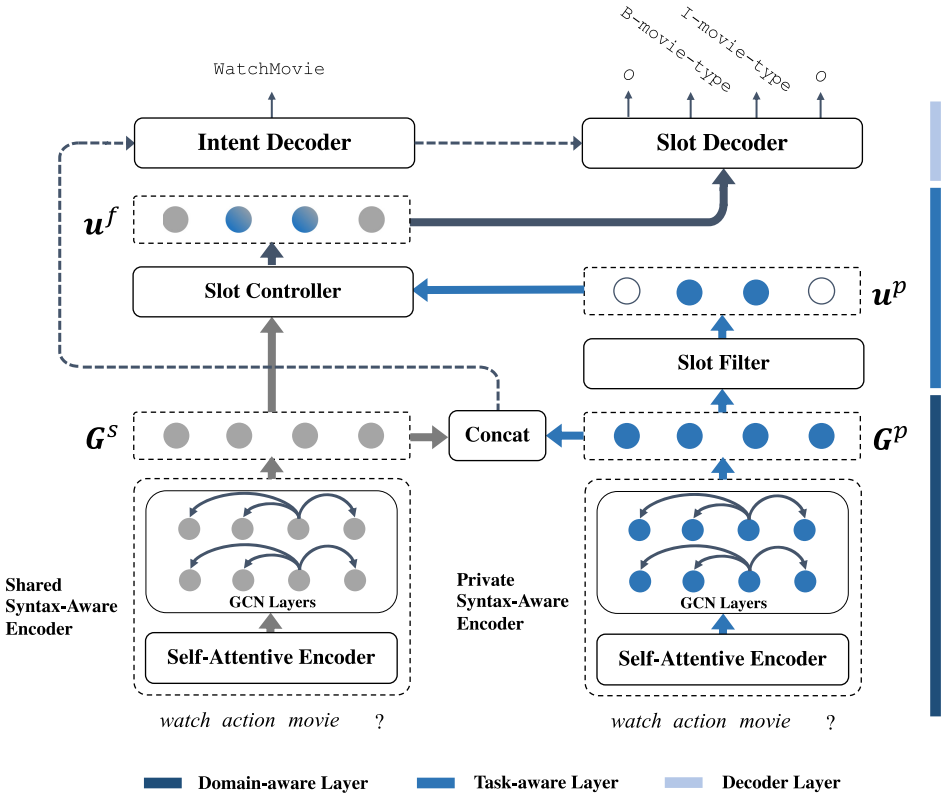


Fig. 3. Overview of our proposed framework. It consists of a shared-private syntax-aware encoder, a slot filter, a slot controller, and two decoders. The gray color represents general features across all domains, and the blue color denotes domain-specific features. For simplicity, we only draw the shared encoder and the encoder of the dth domain in the figure. Better viewed in color.

Slot Filling: **Slot filling (SF)** can be seen as a sequence labeling task to produce a sequence slots $o^S = (o_1^S, \ldots, o_n^S)$, which can be written as:

$$o^S = \texttt{Slot-Filling}(X). \tag{2}$$

Joint Model: Joint model denotes that a joint model predicts the slots sequence and intent simultaneously, which has the advantage of capturing shared knowledge across related tasks, using:

$$(o^I, o^S) = \texttt{Joint-Model}(X). \tag{3}$$

Multi-domain Learning. Suppose that there is a set of domains $D = \{d_1, d_2, \ldots, d_{|D|}\}$ and a dataset with m data instances $T = \{t_1, t_2, \ldots, t_m\}$. For each t in T, we have $t = (X, \mathbf{o}^S, o^I, d)$, where X represents utterance, $\mathbf{o}^S$ represents target slots, o^I represents target intent, and d represents the domain of this data, respectively. The goal is to train a joint model on multiple source domains, which can be used for each domain.

3 APPROACH

The overall structure of our multi-level shared-private framework is shown in Figure 3. First, the model consists of a *shared-private syntactic encoder*, which is used for generating domain-shared

and domain-specific features. Second, the sentence-level shared and private features are combined for intent detection directly. Third, a token-level shared-private framework is used on top of the sentence-level representations, which includes a two-stage decoder and for making fine-grained knowledge transfer for slot filling.

3.1 Shared-private Syntactic Encoder

As shown in Figure 3, a shared syntax-aware encoder is used to capture domain-shared features. Each instance is passed into the shared encoder and its corresponding private encoder to obtain the representation.

Self-attentive Sentence Representation. Following Qin et al. [27], we first use a basic self-attentive encoder to obtain self-attentive representation, which includes a **bidirectional LSTM (BiLSTM)** [11] to obtain the temporal information within words and a self-attention mechanism to capture the contextual information. Given an n-word sentence $X = (x_1, x_2, \ldots, x_n)$, we first use the BiLSTM to read it forwardly from x_1 to x_n and backwardly from x_n to x_1 to produce a series of context-sensitive hidden states $\boldsymbol{H} = \{\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_n\}$, which can be denoted as:

$$\begin{aligned}
\overrightarrow{\boldsymbol{h}_i} &= \overrightarrow{\text{LSTM}}(\phi^{\text{emb}}(x_i), \overrightarrow{\boldsymbol{h}_{i-1}}), i \in [1, n], \\
\overleftarrow{\boldsymbol{h}_i} &= \overleftarrow{\text{LSTM}}(\phi^{\text{emb}}(x_i), \overleftarrow{\boldsymbol{h}_{i+1}}), i \in [n, 1], \\
\boldsymbol{h}_i &= [\overrightarrow{\boldsymbol{h}_i}, \overleftarrow{\boldsymbol{h}_i}],
\end{aligned} \tag{4}$$

where $\phi^{emb}(\cdot)$ denotes the embedding function.

Self-attention is a very effective method of leveraging context-aware features over variable-length sequences for natural language processing tasks [53]. Therefore, we also apply self-attention over word embedding to capture context-aware features. We adopt a Transformer encoder [41], which maps the matrix of input vectors $\boldsymbol{X} = \{\phi^{emb}(x_1), \ldots, \phi^{emb}(x_n)\} \in \mathbb{R}^{n \times d}$ (ϕ^{emb} represents embedding mapping matrix) to queries ($\boldsymbol{Q}$), keys ($\boldsymbol{K}$), and values ($\boldsymbol{V}$) matrices by using different linear projections and output $\boldsymbol{C} \in \mathbb{R}^{T \times d}$ is a weighted sum of values:

$$\boldsymbol{C} = \text{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^{\top}}{\sqrt{d_k}}\right)\boldsymbol{V}, \tag{5}$$

where d_k denotes the dimension of keys. We concatenate these two representations as the self-attentive encoding representation:

$$\boldsymbol{E} = \boldsymbol{H} \oplus \boldsymbol{C}, \tag{6}$$

where $\boldsymbol{E} = (\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_n) \in \mathbb{R}^{n \times 2d}$ and $\oplus$ is concatenation operation.

Graph Convolution over Dependency Trees. Syntax information is an important source of features across domains. We use a GCN [15] over the dependency tree of a sentence[1] to exploit syntactic information. Given a graph with k nodes, an adjacency matrix $\boldsymbol{A} \in \mathbb{R}^{k \times k}$ is used to represent the graph, where $A_{ij} = 1$ if there is an edge going from node i to node j. We denote the lth layer output for node i as $\boldsymbol{g}_i^{(l)}$, where $\boldsymbol{g}_i^{(0)}$ represents the initial state of node i.

Following Zhang et al. [49, 52], we set $\boldsymbol{G}^{(0)} = \boldsymbol{E}$. Given the dependency tree of the input sentence, the graph convolution operated on the node representation can be written as:

$$\boldsymbol{g}_i^{(l)} = \sigma\left(\sum_{j=1}^{n} \tilde{A}_{ij}\boldsymbol{W}^{(l)}\boldsymbol{g}_j^{(l-1)} + \boldsymbol{b}^{(l)}\right), \tag{7}$$

[1]We use Stanford CoreNLP [24] to generate the dependency tree.

where the layer $l \in [1, 2, \ldots, L]$, $\tilde{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{I}$ and $\boldsymbol{I}$ is a $n \times n$ identity matrix to consider information itself, $\boldsymbol{W}^{(l)}$ is a linear transformation, $\boldsymbol{b}^{(l)}$ is a bias term, and σ is a nonlinear function. $\boldsymbol{g}_i^{(L)}$ is the final state of node i.

For shared-private modeling, we allocate a set of parameters Θ shared across all domains, and a set of private parameter Θ_d^p for each domain, $\mathrm{d} \in \{1, \ldots, |D|\}$. Thus, the total set of model parameters is $\Theta \bigcup \Theta_1^p, \ldots, \Theta_D^p$. For testing, given an input utterance, we use Θ^s to calculate a shared representation $\boldsymbol{G}^s$, and Θ_d^p that corresponds to the input domain to calculate a private representation $\boldsymbol{G}^p$.

3.2 Domain-aware Sentence-level Transfer for Intent Detection

We use a standard sentence-level shared-private structure [20, 46] over the input utterance for knowledge transfer concerning intent detection.

Domain Shared-private Feature Fusion. After obtaining the domain-shared and domain-specific encoding representation $\boldsymbol{G}^s$, $\boldsymbol{G}^p$, we use self-attention [4, 54] to aggregate relevant context representation for intent detection:

$$\boldsymbol{a}^s = \boldsymbol{W}^s \boldsymbol{G}^s + \boldsymbol{b}^s, \tag{8}$$

$$\boldsymbol{p}^s = \mathrm{softmax}\left(\boldsymbol{a}^s\right). \tag{9}$$

The shared context representation $\boldsymbol{c}^s$ is computed as the sum of each element $\boldsymbol{g}_i^s$, weighted by the corresponding normalized self-attention score p_i^s:

$$\boldsymbol{c}^s = \sum_i p_i^s \boldsymbol{g}_i^s. \tag{10}$$

We similarly compute the local self-attention context $\boldsymbol{c}^p$.

$\boldsymbol{c}^s$ and $\boldsymbol{c}^p$ are combined by concatenation:

$$\boldsymbol{c}^m = [\boldsymbol{c}^s, \boldsymbol{c}^p], \tag{11}$$

deriving a mixed syntax-aware encoding feature $\boldsymbol{c}^m$.

Intent Prediction. The mixed encoding vector $\boldsymbol{c}^m$ is used as input for intent detection:

$$\boldsymbol{y}^I = \mathrm{softmax}\left(\boldsymbol{W}_h^I \boldsymbol{c}^m\right), \tag{12}$$

$$o^I = \mathrm{argmax}(\boldsymbol{y}^I), \tag{13}$$

where $\boldsymbol{y}^I$ is the output intent distribution; o^I represents the intent label, and $\boldsymbol{W}_h^I$ are trainable parameters of the model.

3.3 Task-aware Token-level Transfer for Slot Filling

A token-level shared-private network is used to model the task-aware token-level transfer for slot filling. Specially, we use a two-stage decoder to consider task characteristics for slot filling, building on top of the shared-private encoder as shown in Figure 3. The first stage uses a *filter* to mask out those domain-general tokens, which do not need domain-specific features,[2] to allow our model to focus on knowledge transfer for more inferable tokens. The second stage makes use of a *controller* module to achieve fine-grained knowledge transfer by automatically calculating the weights for domain-shared and domain-specific features at the token-level.

[2]We treat non-slot labels as domain-general (e.g., slots that are tagged 0).

Slot Filter. We adopt a simple feedforward network as our filter. G^s and G^p are concatenated as input to the filter module:

$$F = \text{sigmoid}\left(W_f[G^s; G^p] + b_f\right), \tag{14}$$

where W_f are trainable parameters, and we define the label $F = (f_1, \ldots, f_n)$ as the probability of domain-general tokens. Correspondingly, $(1 - F) = (1 - f_1), \ldots, (1 - f_n)$ is the probability of domain-specific tokens.

We use the output of the filter module on G^p, where

$$u_i^p = (1 - f_i) \cdot g_i^p. \tag{15}$$

The resulting vectors $U^P = \{u_1^p, \ldots, u_n^p\}$ represent domain-specific features.

Given U^P, we use a controller to generate weights on domain-shared and domain-specific features at the token-level, making a fine-grained fusion between the domain-shared and private features at the token-level.

Slot Controller. We concatenate G^s, G^p and use a simple feedforward network to calculate weights at each token, which can be written as follows:

$$P = \text{sigmoid}\left(W_c[G^s; G^p] + b_c\right). \tag{16}$$

The weights $P = \{p_1, \ldots, p_n\}$ produced by the controller module are used to fuse domain-shared and domain-specific features

$$u_i^f = p_i \cdot u_i^p + (1 - p_i) \cdot g_i^s, \tag{17}$$

where u_i^f is the fused representation at ith token.

Slot Prediction. We use a unidirectional LSTM as the slot-filling decoder. Following Li et al. [17] and Qin et al. [27], we adopt intent information to guide the slot prediction. At the ith decoding step, the decoder state h_i^S can be formalized as:

$$h_i^S = \text{LSTM}\left(h_{i-1}^S, y_{i-1}^S, y^I \oplus u_i^f\right), \tag{18}$$

where h_{i-1}^S is the previous decoder state; y_{i-1}^S is the previous emitted slot label distribution, and y^I is embedding of intent.

Finally, h_i^S is used for slot prediction:

$$y_i^S = \text{softmax}\left(W_h^S h_i^S\right), \tag{19}$$

$$o_i^S = \text{argmax}(y_i^S), \tag{20}$$

where o_i^S is the slot label of the ith word in the utterance.

3.4 Joint Training

We adopt a joint model to consider the two tasks and update parameters in a joint optimization. A cross-entropy loss is used for intent detection:

$$\mathcal{L}_1 \triangleq -\sum_{j=1}^{m} \hat{y}^{j,I} \log\left(y^{j,I}\right). \tag{21}$$

Similarly, the slot-filling objective is:

$$\mathcal{L}_2 \triangleq -\sum_{j=1}^{m} \sum_{i=1}^{n_j} \hat{y}_i^{j,S} \log\left(y_i^{j,S}\right), \tag{22}$$

Table 1. Statistics of Datasets

Dataset	Domains	Train	Dev	Test
MTOD	Reminder, Weather, Alarm	30,521	4,181	8,621
ASMixed	ATIS, SNIPS	17,562	1,200	1,593

where $\hat{\boldsymbol{y}}_j^I$ and $\hat{\boldsymbol{y}}_i^S$ are the gold intent label and gold slot label, respectively; m is the number of training data and n_j is the number of tokens in jth data.

In addition, to further strengthen the filter, we add an auxiliary loss to train the filter as a classification task. The loss function can be denoted as:

$$\mathcal{L}_3 \triangleq -\sum_{j=1}^{m}\sum_{i=1}^{n_j} \hat{y}_i^{j,F} \log\left(f_i^j\right) + \left(1-\hat{y}_i^{j,F}\right)\log\left(1-f_i^j\right), \tag{23}$$

where $\hat{y}_i^{j,F}$ is the gold representation of filter. The final joint objective is formulated as:

$$\mathcal{L}_\theta = \alpha_1\mathcal{L}_1 + \alpha_2\mathcal{L}_2 + \alpha_3\mathcal{L}_3, \tag{24}$$

where α_1, α_2 and α_3 are hyper-parameters.

4 EXPERIMENTS

4.1 Datasets

We conduct experiments on the benchmark MTOD [35].[3] The dataset contains three domains, including Alarm, Reminder, Weather domain. We follow the same format and partition as in Reference [35]. To verify the generalization of the proposed model, we construct another multi-domain SLU dataset (ASMixed) by mixing the ATIS [10] and SNIPS [1] dataset and keeping the train/dev/test partition unchanged. The detailed statistics of the two datasets are shown in Table 1.

4.2 Experimental Settings

The dimensionalities of the embeddings are 64 and of the LSTM hidden units are 256. The dropout ratio is 0.4 and the batch size is 16. The learning rate is 0.001. The GCN layer number is 3 for MTOD and 2 for ASMixed. In the framework, we use Adam [14] to optimize the model parameters and adopt the suggested hyper-parameters.

All experiments are conducted using GeForce RTX 2080Ti GPU. The epoch number is 100 for two datasets, and we do not adopt early stopping strategy. For all experiments, we pick the model that works best on development set and then evaluate it on test set.

Similar to `One-Net` [13], we assume that the input is one utterance without the need to know which domain it comes from. During the test period, we adopt a syntax-aware encoder, which is the same as the shared syntax-aware encoder shown in Figure 3, to directly predict the domain of a given utterance. The result is 99.9% in MTOD dataset and 99.7% in ASMixed dataset. The domain classification is very high due to the explicit features in spoken language utterances, which is consistent with the observation of Gupta et al. [7].

4.3 Baselines

We compare our model with several existing state-of-the-art multi-domain SLU baselines including:

[3]The reason why we do not adopt multiwoz is that multiwoz is mainly proposed to evaluate the dialog state tracking task rather than the spoken language understanding, which makes it hard for us to directly use it as a benchmark for evaluating the multi-domain SLU task.

Table 2. Main Results

	MTOD						ASMixed				
Model	**Overall Exact**	**Slot**	**Intent**	**Reminder Exact**	**Alarm Exact**	**Weather Exact**	**Overall Exact**	**Slot**	**Intent**	**ATIS Exact**	**SNIPS Exact**
Shared-LSTM [9]	88.71	94.87	98.70	82.06	90.19	90.99	76.71	92.55	94.41	81.69	70.41
Separated-LSTM [9]	89.73	94.89	99.01	84.59	89.81	92.18	79.53	92.94	94.79	80.96	77.71
Multi-Domain adv [19]	88.82	94.41	98.87	82.09	88.86	92.05	79.47	91.80	96.48	82.75	75.29
One-Net [13]	89.36	95.25	98.56	83.27	90.15	91.83	78.28	93.38	93.72	81.85	73.80
Locale-agnostic-Universal [16]	88.54	94.16	99.12	81.63	89.58	91.21	79.35	92.10	96.48	82.19	75.71
Coach [23]	89.46	95.13	98.38	83.02	90.69	91.78	81.48	92.87	96.86	82.49	80.20
Ours Framework	**91.27***	**95.69***	**99.20***	**85.62***	**92.37***	**93.29***	**84.81***	**94.30***	**97.30***	**86.53***	**82.62***

Overall Exact, Slot, and Intent Denote the Corresponding Metrics on Whole Datasets, and Domain Exact Represents the Exact Accuracy on Each Domain Separately. The numbers with * indicate that the improvement of our framework overall baselines is statistically significant with $p < 0.05$ under t-test.

(1) `Shared-LSTM` Hakkani-Tür et al. [9] trained a single model for intent detection and slot filling using data from all the domains, which has advantage of incorporating domain-shared knowledge.
(2) `Separated-LSTM`. Hakkani-Tür et al. [9] proposed a single-domain joint model for slot filling and intent detection, which can capture domain-specific features for each domain.
(3) `Multi-Domain Adv`. Liu and Lane [19] applied an adversarial training method for slot filling. For a fair comparison, we add an intent detection module and train the two tasks jointly.
(4) `One-Net`. Kim et al. [13] used data-combined for joint slot filling and intent detection, which is another parameter-shared method to incorporate domain-shared features.
(5) `Locale-agnostic-Universal`. Lee et al. [16] proposed a locale-agnostic universal domain classification model based on multi-task learning.
(6) `Coach`. Liu et al. [23] proposed a coarse-to-fine approach (Coach) for cross-domain slot filling. Besides, slot descriptions are used in the fine stage to help recognize unseen slots, and template regularization is applied to further improve the slot filling performance of similar or the same slot types. This approach achieves the state-of-the-art performance. For a fair comparison, we add the intent detection upon coach model for joint SLU task.

4.4 Overall Results

Following prior work [4, 27], we evaluate the performance of slot filling using F1 score, intent prediction using accuracy, and sentence-level semantic frame parsing using the exact accuracy, measuring the ratio of sentences for which both intent and slot are predicted correctly in a sentence.

Table 2 shows the results of the proposed models on two datasets. We can observe that:

- `Locale-agnostic-Universal` achieves the best performances on intent detection among all baselines, which indicates that explicitly modeling domain-shared and domain-specific is more effective than implicitly capturing shared knowledge with sharing parameters.
- Our model significantly outperforms all the baselines by a large margin and achieves state-of-the-art performance. In particular, on the ASMixed dataset, compared with the best prior joint work `Coach`, we achieve 1.43% improvement on slot-filling task. On the MTOD dataset, the same trend has been witnessed. This indicates the effectiveness of our task-aware token-level shared-private framework, which can effectively transfer fine-grained knowledge for each domain.
- Our framework gains the largest improvements on overall exact. We attribute this to the fact that our proposed domain-aware and task-aware framework can better help transfer domain knowledge between the intent and slots and hence improve the SLU performance.

Table 3. Ablation Experiments on ASMixed

Model	Overall Exact	Slot	Intent	ATIS Exact	SNIPS Exact
Full Model	**84.81**	**94.30**	**97.30**	**86.53**	**82.62**
w/o Multi-level Shared-private Architecture	82.93	93.52	97.24	84.96	80.34
w/o Sentence-level Shared-private Architecture	82.74	93.25	96.92	83.95	81.20
w/o Filter & Controller	82.23	93.28	97.05	83.61	80.48
w/o GCN	83.82	93.82	96.99	85.52	81.62
w/ Oracle	87.26	94.93	96.92	86.76	87.79

4.5 Analysis

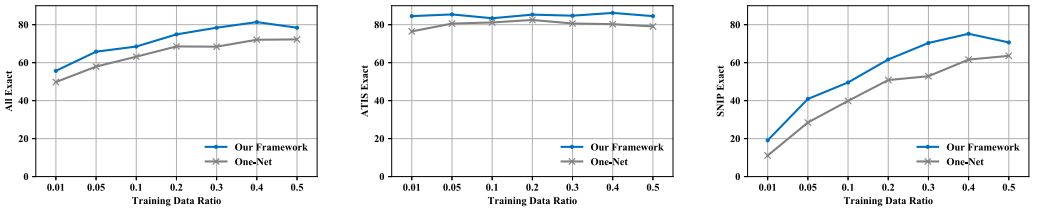
More thorough studies and analysis are conducted on ASMixed in this section, trying to answer the following questions:

(1) Does the domain- and task-aware multi-level shared-private module benefit multi-domain SLU?
(2) Does the domain-aware sentence-level shared-private module benefit multi-domain SLU?
(3) Does the task-aware two-stage decoder successfully transfer fine-grained knowledge for token-level slot filling across all domains?
(4) Can domain-aware syntactic information better generalize across domains in SLU tasks?
(5) Can our framework effectively transfer knowledge for a new domain with little labeled data?
(6) Does our framework successfully capture domain-shared and domain-specific features?
(7) Does our framework still work upon pre-trained model?

4.5.1 Effectiveness of Domain- and Task-aware Multi-level Shared-private Framework. To verify the effectiveness of the proposed domain- and task-aware multi-level shared-private architecture, we conduct a set of experiments where the private syntax-aware encoder is removed. This means that the ablated framework cannot access to domain-specific features. To see the effectiveness of this module fairly, we keep the two-stage module unchanged and use a shared encoder to replace the original private encoder. The results are shown in the *w/o Multi-level Shared-private Architecture* row in Table 3. We observe that the ATIS Exact acc drops by 1.57% and SNIPS Exact acc drops by 2.28% when the shared-private framework is removed. In addition, the whole exact acc drops by 1.88%, which shows that domain-specific feature extracted by the private module is important for multi-domain SLU, including both the sentence-level intent detection subtask and token-level slot filling subtask.

4.5.2 Effectiveness of Domain-aware Sentence-level Shared-private Framework. To verify the effectiveness of the domain-aware sentence-level shared-private architecture, we conduct the experiment where we adopt the shared context representation $\boldsymbol{c}^s$ rather than the mixed shared-private $\boldsymbol{c}^m$ for intent detection and keep other components unchanged. This means that the ablated framework cannot access to domain specific features only for intent detection. The results are shown in the *w/o Sentence-level Shared-private Architecture* row in Table 3. We observe that the intent acc drops by 0.38%, which indicates the sentence-level domain private feature can help intent detection. In addition, the overall exact drops by 2.07%. We attribute it to the reason that intent detection and slot filling are the two correlated tasks where the performance of intent detection affects the whole SLU result.

4.5.3 Effectiveness of Task-aware Token-level Shared-private Framework. To verify the effectiveness of the task-aware token-level shared-private framework, we conduct ablation experiments where we remove the two-stage decoder for slot filling. In this setting, we incorporate shared and domain-specific features by summation other than using our two-stage slot-filling decoder. This



(a) Overall Exact (Acc). (b) ATIS Exact (Acc). (c) SNIPS Exact (Acc)

Fig. 4. Performance (Exact Acc) of domain adaption on different subsets of the original training data on the ASMixed dataset.

model is effectively a domain-aware version of `One-Net` [13], with multi-tasking between intent classification and slot filling only through parameter sharing. The results are shown in the *w/o Filter & Controller* row in Table 3. We can see a 2.58% and a 1.02% drop in the exact and slot-filling metrics, respectively, which verifies the effectiveness of our proposed two-stage decoder. We attribute this to the fact that our filter successfully filters the domain-general token and the model automatically learns weights on how to combine shared and domain-specific features for each token slot prediction.

4.5.4 Oracle Filter Performance. To see the role of our two-stage decoder intuitively, we also present results when using oracle filter information by manually filtering out tokens that are not domain-specific. The results are shown in the oracle row. We obtain 87.26% on overall exact, outperforming our model over 2.45%, which demonstrates better two-stage decoder will lead to better multi-domain SLU performance. The result verifies the effectiveness of our two-stage decoder.

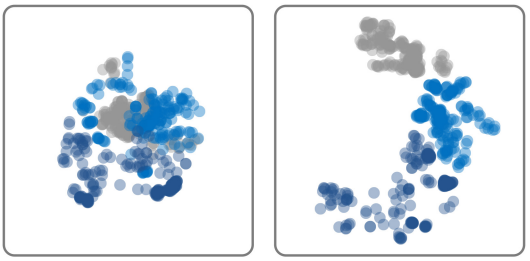
4.5.5 Effectiveness of Domain-aware Syntactic Information. We remove the GCN layers and only adopt the self-attentive encoder to verify the effectiveness of syntax information. The result is shown in the *w/o GCN* row in Table 3. We can see that the performance drops significantly in all metrics, which demonstrates the effectiveness of syntax information in multi-domain SLU tasks. The reason is that utterances in different domains have different syntactic patterns, which helps the model better capture the shared and domain-specific features. To our knowledge, we are the first to show that syntactic information is useful for multi-domain SLU. It is worth noticing that even without the GCN component, our framework still performs the state-of-the-art model [16], which again demonstrates the effectiveness and robustness of our framework.

4.5.6 Domain Adaption. We conduct domain adaption experiments to explore the transferability of our framework on the ASmixed dataset by simulating given a new domain with little labeled data. We keep ATIS dataset unchanged, and the ratio of the other domain SNIPS from the original data varies from [1%, 5%, 10%, 20%, 30%, 40%, 50%]. The results are shown in Figure 4. We can find that our framework outperforms `One-Net` on all ratios of the original dataset. In particular, our framework trained with 20% training dataset can achieve comparable and even better performance compared to `One-Net` with 50% training dataset on some domains. In this case, with 5% training data, our model outperforms `One-Net` by 12.4% on SNIPS exact. This implies that our framework effectively transfers knowledge from other domains to achieve better performance for the low-resources new domain.

4.5.7 Breakdown Evaluation. In this section, we further investigate why our framework is useful in the few-shot setting, where we keep 5% original data as training data. Compared with other domains, we noticed that the slot F1 of the `Weather` domain outperforms `One-Net` in the

Table 4. The Delta of Major Slots' F1 in Weather Domain with 5% Training Data

Model	location	datetime	noun	attribute
One-Net	81.73	90.79	94.68	91.62
Our Framework	**85.46**	**92.28**	**95.67**	**92.24**
Δ	+3.73	+1.49	+0.99	+0.62



(a) (b)

Fig. 5. t-SNE visualization of sentences vector space from the shared encoder (a) and with each domain private encoder (b).

significant range. We conducted a more in-depth analysis of the Weather domain. As shown in Table 4, the slots datetime and location gained the largest improvements. We observe datetime is a domain-shared slot, which occurs in each domain. It shows the architecture successfully transfers the knowledge among different domains to enhance the performance of the model. Moreover, we observe that location is a domain-specific slot that does not have similar slots in other domains, which leads to the difficulty of the prediction of this slot in traditional models. The results show that the shared-private network structure is useful for improving both domain-shared slots and domain-specific slots by its token-level fine-grained knowledge transfer mechanism.

4.5.8 Visualization. To understand whether our framework successfully captures the domain-shared and domain-specific features, we visualize G^s and G^p from the full model trained on MTOD. In particular, we put 600 sentences, which include 200 sentences from the Alarm domain, 200 sentences from the Reminder domain, and 200 sentences from Weather domain, into the shared syntax-aware encoder. Sentences from each domain are fed into its private syntax-aware encoder to get their sentence representations.

We use t-SNE to visualize sentence representations obtained by the shared and private encoders. The vectors are shown in Figure 5. We can observe that those representations from the shared encoder tend to stay closer. In contrast, each private sentence representation from each domain tends to occur in a cluster, and there is nearly no overlap between different domains. This demonstrates our syntax-aware encoders capture the domain-shared and domain-specific features effectively.

4.5.9 Case Study. To better understand how our proposed task-aware two-stage decoder affects and contributes to the final result, we conduct a case study of the slot-filling task between our model and the baseline model One-Net.

This case is shown in Figure 6. For the word "*around*," One-Net predicts its slot label as "O" incorrectly. The token "*around*" is more likely treated as a domain-general token because it usually does not have real meaning in SLU system. The result indicates that One-Net cannot capture sufficient domain information to predict it correctly. In contrast, our model predicts the slot label correctly.

	from	*Indianapolis*	*to*	*Orlando*	*around*	*December*	*twenty*	*fifth*
One-Net	O	From City	O	To City	O	Time Month	Time Day	
Our Model	O	From City	O	To City	Time Relative	Time Month	Time Day	

Fig. 6. Case study. The blue slot is correct, while the red one is wrong. Better viewed in color.

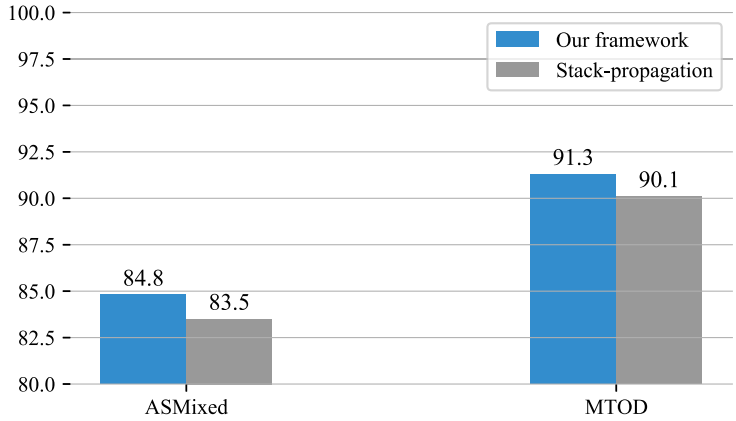


Fig. 7. Performance (Exact Acc) on two datasets between our framework with the SOTA single-domain model (stack-propagation).

We attribute this to the fact that the proposed two-stage decoder successfully learns to capture more domain-specific knowledge for this token, achieving the fine-grained knowledge transfer.

4.5.10 Compared with the Best Single-domain Model. To further verify the effectiveness of our proposed method, we compare our model with the state-of-the-art single-domain model `Stack-propagation` [27]. `Stack-propagation` is directly trained with the mixed dataset, which can be considered as a shared model to implicitly extract domain knowledge.

From the results shown in Figure 7, we can observe that our framework outperforms Stack-propagation on two datasets, which demonstrates that our proposed multi-level shared-private framework makes better domain knowledge representation and transfer than the single-domain-based model.

4.5.11 Effect of RoBERTa. We would like to investigate whether our framwork can still obtain improvement over the pre-trained model (PLMs). To answer the question, we explore the RoBERTa [22] on our framework and we name it as `our framework + RoBERTa`. More specifically, we replace the shared Syntactic Encoder with RoBERTa-base and keep other components unchanged. In the experimental setting, we adopt the fine-tuning mode. For generating token hidden representation, we follow Qin et al. [27] to consider the first subword label if a word is broken into multiple subwords. For example, if sentence "`[<s>]` *The movie is very interesting* `[</s>]`" is split into "`[<s>]` *The movie is very inter## ##esting* `[</s>]`," then we only adopt the representation of *inter* as the whole token *interesting* representation.

The comparison results are shown in Figure 8, we find that our `framework + RoBERTa` outperforms our model on all datasets, which indicates the effectiveness of the pre-trained model.

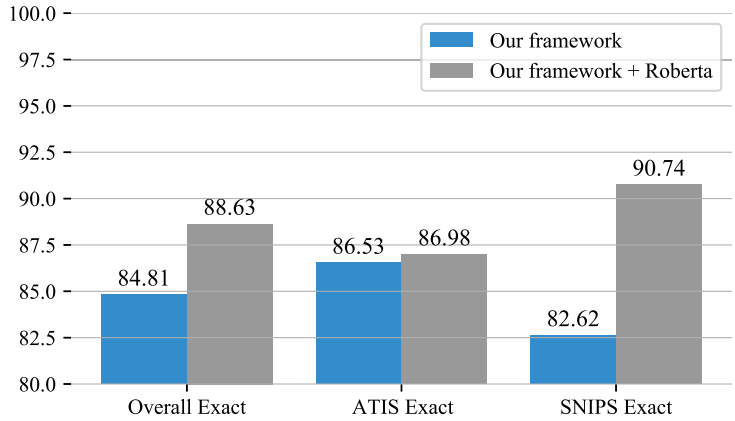


Fig. 8. Performance (Exact Acc) on two datasets between our framework and our framework+RoBERTa.

We attribute this to the fact that pre-trained models can provide rich semantic features, which can help to improve the performance on multi-domain SLU tasks, which has the consistent observation with Qin et al. [26].

5 RELATED WORK

Intent Detection and Slot Filling. Intent detection and slot filling are two core subtasks of **spoken language understanding (SLU)**, which aims to identify users' intents and to extract semantic constituents from the natural language utterances [40]. Intent detection can be considered as the sentence classification task. The classical methods, such as **support vector machine (SVM)** [8] and RNN[34], have been proposed to solve intent detection. Recently, Xia et al. [43] adopts a capsule-based neural network with self-attention for intent detection, achieving the promising performance.

Slot filling can be regarded as a sequence labeling task. The popular approaches are **conditional random fields (CRF)** [33] and **recurrent neural networks (RNN)** [44, 47]. Recently, Qin et al. [27], Shen et al. [36], Tan et al. [38], and Qin et al. [28] propose the self-attention mechanism sequential labeling, which achieves the promising performance without CRF structure.

Joint Model for SLU. Since intent and slots are closely related, dominant methods [4, 17, 27, 29, 31, 39, 42, 50, 51] in the literature adopt the joint model to consider the mutual relationship between slot filling and intent detection. Zhang and Wang [51] proposed a joint model using LSTM for learning the correlation between intent and slots. Goo et al. [4] proposed a slot-gated model to consider the relationship and interaction between two tasks. Li et al. [17] and Qin et al. [27] proposed to use intent information to explicitly model the semantic correlation between slots and intent. However, the above work is restricted to a single domain. In contrast, we consider joint SLU in a multi-domain setting.

Multi-Domain SLU. Hakkani-Tür et al. [9] proposed a single LSTM model over a mixed multi-domain dataset implicitly learning the domain-shared features. Kim et al. [13] adopted one network to jointly modeling slot filling, intent detection, and domain classification. Liu et al. [23] proposed a coarse-to-fine approach (Coach) for cross-domain slot filling. The above methods trained a single model on the mixed dataset. Compared with their work, we propose a domain-aware and task-aware model by extending a shared-private framework into a token-level knowledge sharing structure. In addition, the above works do not incorporate syntax information, and we find that

modeling the syntax information is useful for multi-domain SLU. More closely related to our work, Lee et al. [16] used a shared-private framework for domain classification, and Liu and Lane [19] used a shared-private framework for slot filling. These methods can be regarded as application of the standard shared-private architecture to subproblems in SLU. In contrast to their work, we exploit the mutual benefit between intent detection and slot filling for fine-grained knowledge transfer. To our knowledge, we are the first to investigate shared-private framework for *joint* SLU and the first to conduct the token-level selective weighing shared and private representations in their integration.

Graph Convolutional Network. **Graph convolutional networks (GCN)** are neural networks that operate directly on graph structures [15] to model the structural information, which has been applied successfully in various NLP tasks. De Cao et al. [2] and Lin et al. [18] propose GCN to perform multi-step reasoning on question answering task. Strubell et al. [37] and Marcheggiani and Titov [25] utilize GCN to model the syntactic information for semantic role labeling. Huang and Carley [12] and Zhang et al. [49] apply GCN for aspect-based sentiment classification to consider syntactical constraints. Qin et al. [30, 32] explore the graph network to model the interaction between the slot and multiple intents. Guo et al. [5, 6] successfully propose an attention-guided graph convolutional network to encode the dependency trees for relation extraction. Our work follows the above line of models. We propose to utilize the GCN to explore the graph structure to encode the syntactic information for multi-domain SLU tasks. To the best of our knowledge, we are the first to incorporate syntactic information with GCN for multi-domain SLU.

6 CONCLUSION

We investigated domain-aware and task-aware parameterization for multi-domain SLU by building a model with separate domain- and task-specific parameters. In particular, a domain-aware sentence-level shared-private framework can be used for extracting domain knowledge for intent detection, while a task-aware token-level shared-private framework is used to achieve a fine-grained knowledge transfer for slot filling. Unlike existing methods, which use the same parameters for multi-task learning, our model can achieve a fine-grained combination of domain knowledge transfer. Experiments on two publicly available datasets with five domains show the effectiveness of the proposed models, and we achieve state-of-the-art performance. In addition, our model can quickly adapt to a new domain given little labeled data, which makes it more robust and scalable in the real-world scenario. Finally, our work is the first attempt to explore syntax information and empirically demonstrate the effectiveness of syntax information in multi-domain SLU tasks.

REFERENCES

[1] Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril et al. 2018. Snips voice platform: An embedded spoken language understanding system for private-by-design voice interfaces. *arXiv preprint arXiv:1805.10190* (2018).

[2] Nicola De Cao, Wilker Aziz, and Ivan Titov. 2019. Question answering by reasoning across documents with graph convolutional networks. In *Proceedings of the NAACL*. Association for Computational Linguistics, 2306–2317. DOI: https://doi.org/10.18653/v1/N19-1240

[3] Haihong E. Peiqing Niu, Zhongfu Chen, and Meina Song. 2019. A novel bi-directional interrelated model for joint intent detection and slot filling. In *Proceedings of the ACL*. Association for Computational Linguistics, 5467–5471. DOI: https://doi.org/10.18653/v1/P19-1544

[4] Chih-Wen Goo, Guang Gao, Yun-Kai Hsu, Chih-Li Huo, Tsung-Chieh Chen, Keng-Wei Hsu, and Yun-Nung Chen. 2018. Slot-gated modeling for joint slot filling and intent prediction. In *Proceedings of the NAACL*. Association for Computational Linguistics, 753–757. DOI: https://doi.org/10.18653/v1/N18-2118

[5] Zhijiang Guo, Yan Zhang, and Wei Lu. 2019. Attention guided graph convolutional networks for relation extraction. In *Proceedings of the ACL*. Association for Computational Linguistics, 241–251. DOI: https://doi.org/10.18653/v1/P19-1024

[6] Zhijiang Guo, Yan Zhang, Zhiyang Teng, and Wei Lu. 2019. Densely connected graph convolutional networks for graph-to-sequence learning. *Trans. Assoc. Computat. Ling.* 7 (Mar. 2019), 297–312. DOI: https://doi.org/10.1162/tacl_a_00269

[7] Raghav Gupta, Abhinav Rastogi, and Dilek Hakkani-Tur. 2018. An efficient approach to encoding context for spoken language understanding. *arXiv preprint arXiv:1807.00267* (2018).

[8] Patrick Haffner, Gokhan Tur, and Jerry H. Wright. 2003. Optimizing SVMs for complex call classification. In *Proceedings of the ICASSP*.

[9] Dilek Hakkani-Tür, Gökhan Tür, Asli Celikyilmaz, Yun-Nung Chen, Jianfeng Gao, Li Deng, and Ye-Yi Wang. 2016. Multi-domain joint semantic frame parsing using bi-directional RNN-LSTM. In *Proceedings of the Interspeech*.

[10] Charles T. Hemphill, John J. Godfrey, and George R. Doddington. 1990. The ATIS spoken language systems pilot corpus. In *Proceedings of the Workshop on Speech and Natural Language*.

[11] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computat.* 9, 8 (1997).

[12] Binxuan Huang and Kathleen Carley. 2019. Syntax-aware aspect level sentiment classification with graph attention networks. In *Proceedings of the EMNLP*.

[13] Young-Bum Kim, Sungjin Lee, and Karl Stratos. 2017. OneNet: Joint domain, intent, slot prediction for spoken language understanding. In *Proceedings of the ASRU*. IEEE, 547–553.

[14] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[15] Thomas N. Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[16] Jihwan Lee, Ruhi Sarikaya, and Young-Bum Kim. 2019. Locale-agnostic universal domain classification model in spoken language understanding. In *Proceedings of the NAACL*. Association for Computational Linguistics.

[17] Changliang Li, Liang Li, and Ji Qi. 2018. A self-attentive model with gate mechanism for spoken language understanding. In *Proceedings of the EMNLP*. Association for Computational Linguistics, 3824–3833. DOI: https://doi.org/10.18653/v1/D18-1417

[18] Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. 2019. KagNet: Knowledge-aware graph networks for commonsense reasoning. In *Proceedings of the EMNLP*. Association for Computational Linguistics, 2829–2839. DOI: https://doi.org/10.18653/v1/D19-1282

[19] Bing Liu and Ian Lane. 2017. Multi-domain adversarial learning for slot filling in spoken language understanding. *arXiv preprint arXiv:1711.11310* (2017).

[20] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2017. Adversarial multi-task learning for text classification. In *Proceedings of the ACL*.

[21] Yijin Liu, Fandong Meng, Jinchao Zhang, Jie Zhou, Yufeng Chen, and Jinan Xu. 2019. CM-Net: A novel collaborative memory network for spoken language understanding. In *Proceedings of the EMNLP*. Association for Computational Linguistics, 1051–1060. DOI: https://doi.org/10.18653/v1/D19-1097

[22] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).

[23] Zihan Liu, Genta Indra Winata, Peng Xu, and Pascale Fung. 2020. Coach: A coarse-to-fine approach for cross-domain slot filling. arXiv:cs.CL/2004.11727 (2020).

[24] Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of the ACL*.

[25] Diego Marcheggiani and Ivan Titov. 2017. Encoding sentences with graph convolutional networks for semantic role labeling. In *Proceedings of the EMNLP*. Association for Computational Linguistics, 1506–1515. DOI: https://doi.org/10.18653/v1/D17-1159

[26] Libo Qin, Wanxiang Che, Yangming Li, Minheng Ni, and Ting Liu. 2020. DCR-Net: A deep co-interactive relation network for joint dialog act recognition and sentiment classification. In *AAAI*. 8665–8672.

[27] Libo Qin, Wanxiang Che, Yangming Li, Haoyang Wen, and Ting Liu. 2019. A stack-propagation framework with token-level intent detection for spoken language understanding. In *Proceedings of the EMNLP*. Association for Computational Linguistics, 2078–2087. DOI: https://doi.org/10.18653/v1/D19-1214

[28] Libo Qin, Tailu Liu, Wanxiang Che, Bingbing Kang, Sendong Zhao, and Ting Liu. 2020. A co-interactive transformer for joint slot filling and intent detection. *arXiv preprint arXiv:2010.03880* (2020).

[29] Libo Qin, Minheng Ni, Yue Zhang, and Wanxiang Che. 2020. CoSDA-ML: Multi-lingual code-switching data augmentation for zero-shot cross-lingual NLP. arXiv:cs.CL/2006.06402 (2020).

[30] Libo Qin, Fuxuan Wei, Tianbao Xie, Xiao Xu, Wanxiang Che, and Ting Liu. 2021. GL-GIN: Fast and accurate non-autoregressive model for joint multiple intent detection and slot filling. In *Proceedings of the ACL*.

[31] Libo Qin, Tianbao Xie, Wanxiang Che, and Ting Liu. 2021. A survey on spoken language understanding: Recent advances and new frontiers. arXiv:cs.CL/2103.03095 (2021).
[32] Libo Qin, Xiao Xu, Wanxiang Che, and Ting Liu. 2020. AGIF: An adaptive graph-interactive framework for joint multiple intent detection and slot filling. In *Proceedings of the EMNLP*. Association for Computational Linguistics, 1807–1816. DOI: https://doi.org/10.18653/v1/2020.findings-emnlp.163
[33] Christian Raymond and Giuseppe Riccardi. 2007. Generative and discriminative algorithms for spoken language understanding. In *Proceedings of the Interspeech*.
[34] Ruhi Sarikaya, Geoffrey E. Hinton, and Bhuvana Ramabhadran. 2011. Deep belief nets for natural language call-routing. In *Proceedings of the ICASSP*.
[35] Sebastian Schuster, Sonal Gupta, Rushin Shah, and Mike Lewis. 2019. Cross-lingual transfer learning for multilingual task oriented dialog. In *Proceedings of the NAACL*. Association for Computational Linguistics, 3795–3805. DOI: https://doi.org/10.18653/v1/N19-1380
[36] Tao Shen, Tianyi Zhou, Guodong Long, Jing Jiang, Shirui Pan, and Chengqi Zhang. 2018. DiSAN: Directional self-attention network for RNN/CNN-free language understanding. In *Proceedings of the AAAI*.
[37] Emma Strubell, Patrick Verga, Daniel Andor, David Weiss, and Andrew McCallum. 2018. Linguistically informed self-attention for semantic role labeling. *arXiv preprint arXiv:1804.08199* (2018).
[38] Zhixing Tan, Mingxuan Wang, Jun Xie, Yidong Chen, and Xiaodong Shi. 2018. Deep semantic role labeling with self-attention. In *Proceedings of the AAAI*.
[39] Dechuang Teng, Libo Qin, Wanxiang Che, Sendong Zhao, and Ting Liu. 2021. Injecting word information with multi-level word adapter for Chinese spoken language understanding. arXiv:cs.CL/2010.03903 (2021).
[40] Gokhan Tur and Renato De Mori. 2011. *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech*. John Wiley & Sons.
[41] 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 *Proceedings of the NIPS*.
[42] Yu Wang, Yilin Shen, and Hongxia Jin. 2018. A bi-model based RNN semantic frame parsing model for intent detection and slot filling. In *Proceedings of the ACL*.
[43] Congying Xia, Chenwei Zhang, Xiaohui Yan, Yi Chang, and Philip Yu. 2018. Zero-shot user intent detection via capsule neural networks. In *Proceedings of the EMNLP*. Association for Computational Linguistics, 3090–3099. DOI: https://doi.org/10.18653/v1/D18-1348
[44] Puyang Xu and Ruhi Sarikaya. 2013. Convolutional neural network based triangular CRF for joint intent detection and slot filling. In *Proceedings of the ASRU*.
[45] Puyang Xu and Ruhi Sarikaya. 2013. Exploiting shared information for multi-intent natural language sentence classification. In *Proceedings of the Interspeech*.
[46] Zhilin Yang, Ruslan Salakhutdinov, and William Cohen. 2016. Multi-task cross-lingual sequence tagging from scratch. *arXiv preprint arXiv:1603.06270* (2016).
[47] Kaisheng Yao, Baolin Peng, Yu Zhang, Dong Yu, Geoffrey Zweig, and Yangyang Shi. 2014. Spoken language understanding using long short-term memory neural networks. In *Proceedings of the SLT*.
[48] Steve Young, Milica Gašić, Blaise Thomson, and Jason D. Williams. 2013. POMDP-based statistical spoken dialog systems: A review. In *Proceedings of the IEEE*. DOI: 10.1109/JPROC.2012.2225812
[49] Chen Zhang, Qiuchi Li, and Dawei Song. 2019. Aspect-based sentiment classification with aspect-specific graph convolutional networks. In *Proceedings of the EMNLP*.
[50] Chenwei Zhang, Yaliang Li, Nan Du, Wei Fan, and Philip Yu. 2019. Joint slot filling and intent detection via capsule neural networks. In *Proceedings of the ACL*.
[51] Xiaodong Zhang and Houfeng Wang. 2016. A joint model of intent determination and slot filling for spoken language understanding. In *Proceedings of the IJCAI*.
[52] Yuhao Zhang, Peng Qi, and Christopher D. Manning. 2018. Graph convolution over pruned dependency trees improves relation extraction. In *Proceedings of the EMNLP*.
[53] Victor Zhong, Caiming Xiong, and Richard Socher. 2018. Global-locally self-attentive encoder for dialogue state tracking. In *Proceedings of the ACL*. Association for Computational Linguistics, 1458–1467. DOI: https://doi.org/10.18653/v1/P18-1135
[54] Victor Zhong, Caiming Xiong, and Richard Socher. 2018. Global-locally self-attentive encoder for dialogue state tracking. In *Proceedings of the ACL*.

Received April 2020; revised September 2021; accepted November 2021