A DYNAMIC GRAPH INTERACTIVE FRAMEWORK WITH LABEL-SEMANTIC INJECTION FOR SPOKEN LANGUAGE UNDERSTANDING

Zhihong Zhu\(^1\), Weiyuan Xu\(^1\), Xuxin Cheng\(^1\), Tengtao Song\(^1\), Yuexian Zou\(^1,2,\ast\)

\(^1\)ADSPLAB, School of ECE, Peking University \hspace{1em} \(^2\)Peng Cheng Laboratory
\{zhihongzhu, xuwy, chengxx, songtengtao\}@stu.pku.edu.cn, zouyx@pku.edu.cn

ABSTRACT

Multi-intent detection and slot filling joint models are gaining increasing traction since they are closer to complicated real-world scenarios. However, existing approaches (1) focus on identifying implicit correlations between utterances and one-hot encoded labels in both tasks while ignoring explicit label characteristics; (2) directly incorporate multi-intent information for each token, which could lead to incorrect slot prediction due to the introduction of irrelevant intent. In this paper, we propose a framework termed DGIF, which first leverages the semantic information of labels to give the model additional signals and enriched priors. Then, a multi-grain interactive graph is constructed to model correlations between intents and slots. Specifically, we propose a novel approach to construct the interactive graph based on the injection of label semantics, which can automatically update the graph to better alleviate error propagation. Experimental results show that our framework significantly outperforms existing approaches, obtaining a relative improvement of 6.5\% over the previous best model on the MixATIS dataset in overall accuracy.

Index Terms— Spoken Language Understanding, Multi-intent Classification, Slot Filling, Multitask Learning

1. INTRODUCTION

Spoken language understanding (SLU) is a crucial component in task-oriented dialogue systems \([1,2]\), which typically consists of two subtasks: intent detection (ID) and slot filling (SF). As described by \([3]\), complicated real-world scenarios frequently involve multiple intents in a single utterance. Take an example in Fig.1, the task of ID should classify both intent labels in the utterance (i.e., AddToPlaylist and PlayMusic), while SF can be treated as a sequence labeling task to predict slot for each token in BIO format \([4]\).

Since intents and slots are inextricably related \([5,6,7,8]\), researchers in recent years \([9,10,11]\) have increasingly focused on joint multi-intent detection and slot filling. Although achieving promising performance, existing approaches typically classify an utterance to intents represented by one-hot encoding (e.g., 0) while the same problem occurs in slot filling. They ignore intuitive and explicit label characteristics, oversimplifying representations of labels. We argue that the label semantics may be useful, which could improve performances for both subtasks by assessing semantic similarity between words in utterances and words in labels.

Another key challenge in multi-intent SLU is how to effectively incorporate multiple intents information to guide the slot prediction. To handle this, \([9]\) first investigated a multitask network with a slot-gated mechanism \([12]\). For fine-grained interaction between multiple intents and slots, \([10]\) proposed an adaptive graph interactive framework, which builds an interactive graph for each token in the utterance by using all predicted intents. \([11]\) further explored a globally graph interaction network, which models slot dependency and intent-slot interaction for each utterance. However, different tokens appearing in the utterance have various importance for representing the intents. Unfortunately, models mentioned above straightforwardly attach multiple intent information to all tokens, including those without contribution to intent representations, which will introduce noise into sentence-level semantics to some extent.

In this paper, we propose a novel framework DGIF for joint multiple ID and SF. Inspired by the success of leveraging label characteristics to help model optimization \([13,14,15]\), we construct intent and slot spaces using words in each intent label and slot label respectively to inject label information into utterance representations adaptively. Moreover, we propose label-aware regularization to model rich semantic dependencies among labels in each label space. Then, we capture relevant intents for each token to construct the multi-grain intent-slot interactive graph, as opposed to prior works which directly incorporate multiple intents information stat-

---

This paper was partially supported by Shenzhen Science & Technology Research Program (No.GXWD20201231165807007-20200814115301001) and NSFC (No: 62176008).

\ast Corresponding author: zouyx@pku.edu.cn.

\(^{1}\)Code is available at https://github.com/Zhihong-Zhu/DGIF.

![Fig. 1: Utterance with multiple intents (red) and slots (blue).](image-url)
Fig. 2: The architecture of DGF. Two BERT encoders will encode both the utterance (2.1) and labels (2.2). Sentence-level representation \( r^n \) and token-level representation \( h_n \) will be injected with label semantics in respective space (2.3) and interact in 2.4. ✓ denotes the label is selected as prediction. For simplicity, we only draw one case with several labels.

2. METHODOLOGY

As shown in Fig. 2, our framework consists of four major components, and we use a joint training scheme to optimize multi-intent detection and slot filling simultaneously.

2.1. Utterance Encoder

For the input utterance \( U = (u_1, \ldots, u_n) \), we first prepend [CLS] and appended [SEP], in order to match the input of BERT [10]. Then, we employ a self-attentive network [17] over the output \( h = (h_{CLS}, h_1, \ldots, h_n, h_{SEP}) \) of BERT’s encoder to capture the sentence representation \( r^n \) with context-aware features, where \( h \in \mathbb{R}^{(n+2) \times d} \) and \( r^n \in \mathbb{R}^d \).

2.2. Label Embedder

We apply two steps to get label representations. Concretely, considering the ambiguity of label semantics, we first manually convert slot label names to their natural language forms (e.g., “B-PER” to “begin person”) while maintaining intent label names. Then, another BERT with self-attentive layer (2.1) is adopted to obtain label representation \( r^\phi_i \), where \( \phi \in \{I, S\} \) (\( I \) denotes intent label and \( S \) denotes slot label). The reason to use a different BERT is that the utterance and labels commonly differ in syntactic structure.

2.3. Adaptive Label-Semantic Injection Layer

Label-Semantic Injection. We leverage best approximation [18] idea to help incorporate label information into utterance representations. The best approximation problem specifies that \( T \) is a subspace of Hilbert space \( S \). For a given vector \( x \in S \), we need to find the closest point \( \hat{x} \in T \). It turns out solution of \( \hat{x} = \sum_{n=1}^{N} a_n v_n \) will be a linear combination of a basis \( v_1, \ldots, v_N \) for \( T \) of \( N \) dimension. Coefficients \( a_n \) satisfies \( G a = b \) where \( b_n = \langle x, v_n \rangle \) and \( G_{k,n} = \langle v_n, v_k \rangle \).

Instead of directly utilizing the \( r^n \) and \( h \) to predict the intent and slot labels, we first construct the label space \( T^\phi \) with a basis \( \{r^\phi_i, \ldots, r^\phi_{|\phi|}\} \) by \( \{\phi\} \) label embeddings in (2.2). Then for a given \( r^n \), we can project it onto \( T^\phi \) to obtain its best approximation \( \hat{r}^\phi = w^T r^n \), where \( w^T = G^T b \).

The Gram matrix \( G^T \) and \( b \) are calculated as follows:

\[
G^T = \begin{bmatrix}
\langle r^\phi_i, r^\phi_i \rangle & \ldots & \langle r^\phi_i, r^\phi_{|\phi|} \rangle \\
\vdots & \ddots & \vdots \\
\langle r^\phi_{|\phi|}, r^\phi_i \rangle & \ldots & \langle r^\phi_{|\phi|}, r^\phi_{|\phi|} \rangle
\end{bmatrix},
\quad b = \begin{bmatrix}
\langle r^n, r^\phi_i \rangle \\
\vdots \\
\langle r^n, r^\phi_{|\phi|} \rangle
\end{bmatrix}.
\tag{1}
\]

Similarly, \( \hat{h}^\phi_i \) can be derived like Eq. [1]

Label-Aware Regularization. While the label-semantic injection dynamically injects label information into utterance representations, we argue that it ignores the semantic dependencies among labels. Thus, we propose label-aware regularization to ensure that the projected representation \( \hat{r}^\phi \) (\( h_n \)) capture the topological structure of the label space \( T^\phi \) well. Concretely, we use Euclidean and Cosine distance to measure similarities, and optimize label representations as follows:

\[
L_{\text{inter}}^\phi = 1 + \frac{1}{Q|\phi|} \sum_{i=1}^{|\phi|} \sum_{j=1}^Q \cos(r^\phi_i, r^\phi_j),
\quad L_{\text{intra}}^\phi = \frac{1}{PQ} \sum_{i=1}^P \sum_{j=1}^Q |r^n - r^\phi_i|^2,
\]

where \( P \) is the number of samples (length of \( U \)), \( Q \) is the number of gold labels in \( r^n \) (\( h_n \)). The regularization loss of \( T^\phi \) is \( L_{\text{RE}}^\phi = L_{\text{inter}}^\phi + \lambda \times L_{\text{intra}}^\phi \) and \( \lambda \) is a hyper-parameter.

Multiple Intent Detection Decoder. Following [10, 11], after obtaining \( F^n \), we can use it for multi-intent detection:

\[
p^T = \sigma(W_I(\text{LeakyReLU}(W_u \hat{r}^\phi + b_u) + b_i)),
\]

where \( p^T \) denotes intent probability values, \( \sigma \) denotes the sigmoid activation function. \( W_u \) and \( b_u \) are trainable matrix parameters. Then, we apply the same mechanism as [19] to obtain intent number \( O^{\text{IND}} \) of the input utterance:

\[
O^{\text{IND}} = \arg\max(\text{softmax}(W_{\text{ind}} h_{CLS} + b_{\text{ind}})),
\]

where \( W_{\text{ind}} \) is a trainable matrix parameter. After that, we
choose the top $O_{\text{IND}}$ in $p^t$ as the final intent result $O^t = \{o^t_1, \ldots, o^t_{|O_{\text{IND}}|}\}$.

2.4. Dynamic Graph Interaction Layer

Graph Construction. Mathematically, our graph can be denoted as $G = (V, E)$ where vertices refer to intents and slots, edges refer to correlations between them.

Vertices. We have $n + m$ number of nodes in the interactive graph where $n$ is the utterance length and $m$ is the number of intents $O_{\text{IND}}$ in §2.3. The input of slot token feature is $G_{|S|} = \{h_i\}_1^n$ while the input of intent feature is $G_{|I|} = \{\phi^{emb}(o'_i), \ldots, \phi^{emb}(o'_{|O_{\text{IND}}|})\}$ where $\phi^{emb}$ is the embedding mapping function to map $o'_i$ to its embedding $r^i$.

The first layer states vector for two kind of nodes is $G_1 = \{G_{|I|}, G_{|S|} = \{\phi^{emb}(o'_i), \ldots, \phi^{emb}(o'_{|O_{\text{IND}}|}), h_1, \ldots, h_n\}$.

Edges. There are three types of connections in the graph:

(a) ‘intent’-’intent’ connection: We connect all predicted intent labels to each other to model the relationship between them, since all of them appear in the same utterance.

(b) ‘slot’-’slot’ connection: We connect each slot token to other slot tokens with a window size to model slot dependency and incorporate bidirectional contextual information.

(c) ‘intent’-’slot’ connection: We adopt a scaled dot-product attention mechanism [20] for computing relevance between intent embedding and slot embedding as follows:

$$\delta_{ij} = \frac{\exp(h_i \phi^{emb}(o'_j)^\top / \sqrt{d})}{\sum_{k=1}^n \exp(h_i \phi^{emb}(o'_k)^\top / \sqrt{d})}$$

where $d$ is the dimension of hidden states, $\delta_{ij}$ is the relevance score between $i$-th token and $j$-th intent. We innovatively employ a hyper-parameter $\delta$ to measure intent and token relevance. If $\delta_{ij} > \delta$, it indicates that the token plays a significant role in determining the intent. In this case, this token is directly connected to the relevant intent (cf. Fig 4 right).

Graph Network. We use Graph Attention Network (GAT) [21] to model intent-slot interaction. Specifically, for a given graph with $n$ nodes, GAT take the initial node features $H = \{h_1, \ldots, h_n\}$ as input to produce more abstract representation $H' = \{h'_1, \ldots, h'_n\}$ as its output. Within the graph, the aggregation at $l$-th layer can be defined as:

$$g_{i}^{S,\text{L}+1} = \sigma(\sum_{j \in G^S} \alpha_{ij}W_S g_{j}^{S,l} + \sum_{j \in G^l} \alpha_{ij}W_S g_{j}^{l,d}),$$

where $\alpha_{ij}$ is the attention coefficient, $G^S$ and $G^l$ are vertices sets which denotes connected slots and intents, respectively.

Slot Filling Decoder. After $L$ layers’ propagation, we obtain the final slot representation $G_{|S|,L+1}$ for slot prediction:

$$O^t_S = \text{argmax}(\text{softmax}(W_S g_{t}^{S,L+1} + b_S)),$$

in which $W_S$ is a trainable parameter and $O^t_S$ is the predicted slot of the $t$-th token in an utterance.

2.5. Joint Training

We adopt joint training to learn parameters. Multi-intent ($L_{\text{ID}}$) and its number ($L_{\text{IND}}$) detection are trained with binary cross-entropy while slot filling ($L_{\text{SF}}$) is trained with cross-entropy. The final joint objective is formulated as:

$$\mathcal{L} = \alpha(L_{\text{ID}} + \gamma L_{\text{RE}}^I) + \beta(L_{\text{SF}} + \gamma L_{\text{RE}}^S) + (1 - \alpha)L_{\text{IND}},$$

where $\alpha$, $\beta$ and $\gamma$ are trade-off hyper-parameters.

3. EXPERIMENTS

3.1. Datasets and Metrics

Datasets. We conduct experiments on two multi-intent SLU benchmark:\footnote{https://github.com/LooperXX/AGIF} MixATIS [11] is constructed from ATIS [22] and generated by concatenating single utterances with the conjunctions, e.g., “and”. MixATIS containing 13,162/ 756/ 828 utterances for train/ validation/ test. MixSNIPS [11,23] contains 39,776/ 2,198/ 2,199 utterances for train/ validation/ test. In addition, both of datasets are the cleaned version.

Metrics. We evaluate performance using F1 score of slot filling (Slot F1), accuracy of intent detection (Intent Acc) and sentence-level overall accuracy (Overall Acc) as in [11,12]. “Overall Acc” considers prediction of an utterance is correct only when its intents and slots are all correctly predicted.

3.2. Experimental Settings

Considering the inference speed [24], we use English uncased BERT-Base model [16] which consists of 12 layers, 12 heads and 768 hidden states. The batch size is 32 and the epoch is 50. Adam is used for optimization with learning rate of 2e-5. The layer number of GAT is set to 2. For hyper-parameters of loss $\alpha$, $\beta$ and $\gamma$ are empirically set as 0.6: 1: 0.3 for MixATIS and 0.8: 1.2: 0.2 for MixSNIPS.

3.3. Baselines

We compare our DGIF with both single-intent and multi-intent models. For single-intent SLU models to handle multi-intent utterances, multiple intents are concatenated with ‘#’ into a single intent for a fair comparison. We also combine pre-trained language models (PLMs) with competitive baselines for comparison which are AGIF/GL-GIN + Bert-base. Following [30], we obtain the hidden state of the first special token ([CLS]) with sigmoid function for detecting multi-intent and use hidden states of utterance tokens for slot filling.

3.4. Main Results

Table 1 shows experiment results of different models on the MixATIS and MixSNIPS datasets. From the results, we have the following observations:

(1) For slot FI, our method leads to slight improvements compared to the best baseline SLIM, which validates that DGIF is effective on slot filling. (2) Turning to intent accuracy, DGIF exceeds SLIM by 5.0% and 0.6%, respectively. It proves that DGIF has a strong ability to identify intents. (3) Moreover, DGIF surpasses SLIM 3.1% and 0.3% on overall accuracy, which confirms that DGIF is more powerful in understanding the implicit correlations between intents and slots.
Effectiveness of Adaptive Label-Semantic Injection Layer.

(1) We remove label-aware regularization, which is named as w/o LAR in Table 2. From the results, we can observe the absence of label-aware regularization leads to 0.6% slot F1 and 2.1% intent accuracy drops. This indicates that the label-aware regularization encourages our network to model rich semantic dependencies among labels in each label space.

(2) We further remove label-semantic injection and utilizes the product of utterance/token representations and labels and keep other components unchanged. It is named as w/o LAR + LSI in Table 2. We can obviously observe that the overall accuracy drops by 3.4%. This indicates that label-semantic injection can capture the correlation between utterances and explicit labels’ semantics, which is beneficial for the semantic performance of multi-intent SLU system.

Effectiveness of Dynamic Graph Interaction Layer. On the basis of the previous experiments, we replace the dynamic interactive layer with the vanilla attention mechanism, which is named as w/o LAR + LSI + GIL in Table 2. We can observe the performance drops in all metrics on the MixATIS dataset. We attribute it to the fact that our approach can automatically filter irrelevant intent information for each token.

Visualization. To better understand what the interactive graph layer has learned, we visualize the attention weight of it and SDJN [28] counterpart version for comparison, which is shown in Fig. 3. We find that DGIF properly aggregates relevant intent “GetWeather” at slots “forecast” and “cold” respectively where the attention weights successfully focus on the correct slot. This justifies our DGIF has a better interaction ability compared to the prior approach.

Table 1: Main results on two multi-intent datasets. ♦: Single-intent SLU models. ♣: Multi-intent SLU models. SDJN [28]’s code is not released. The best performance is in bold and the second best performance is underlined. † indicate that the improvement of DGIF over all baselines is statistically significant with p < 0.05 under t-test.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Slot (F1)</td>
<td>Intent (Acc)</td>
</tr>
<tr>
<td>SF-ID♦ 19</td>
<td>BiLSTM</td>
<td>87.4</td>
<td>66.2</td>
</tr>
<tr>
<td>Stack-Propagation♦ 19 [27]</td>
<td>Self-attentive [27]</td>
<td>87.8</td>
<td>72.1</td>
</tr>
<tr>
<td>Joint Multiple ID-SF♦ 19 [9]</td>
<td>BiLSTM</td>
<td>84.6</td>
<td>73.4</td>
</tr>
<tr>
<td>LR-Transformer♦ 21</td>
<td>Transformer [20]</td>
<td>88.0</td>
<td>76.1</td>
</tr>
<tr>
<td>AGIF♣ 20 [18]</td>
<td>Self-attentive [27]</td>
<td>86.7</td>
<td>74.4</td>
</tr>
<tr>
<td>+ BERT</td>
<td>BERT [16]</td>
<td>87.5</td>
<td>77.2</td>
</tr>
<tr>
<td>GL-GIN♣ 21 [19]</td>
<td>Self-attentive [27]</td>
<td>88.3</td>
<td>76.3</td>
</tr>
<tr>
<td>+ BERT</td>
<td>BERT [16]</td>
<td>88.4</td>
<td>78.0</td>
</tr>
<tr>
<td>SDJN♣ 22 [11]</td>
<td>Self-attentive [27]</td>
<td>88.2</td>
<td>77.1</td>
</tr>
<tr>
<td>SLIM♣ 22 [11]</td>
<td>BERT [16]</td>
<td>88.5</td>
<td>78.3</td>
</tr>
</tbody>
</table>

DGIF (Ours) BERT [16] 88.5 83.3† 50.7† 95.9 97.8† 84.3†

Table 2: Results of ablation test on the MixATIS dataset.

<table>
<thead>
<tr>
<th>Variant</th>
<th>Dataset: MixATIS [22][11]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Slot</td>
</tr>
<tr>
<td></td>
<td>(F1)</td>
</tr>
<tr>
<td>DGIF (Ours)</td>
<td>88.5</td>
</tr>
<tr>
<td>w/o LAR</td>
<td>87.9 (0.6)</td>
</tr>
<tr>
<td>w/o LAR + LSI</td>
<td>87.1 (1.4)</td>
</tr>
<tr>
<td>w/o LAR + LSI + GIL</td>
<td>86.4 (2.1)</td>
</tr>
</tbody>
</table>

Fig. 3: Attention heatmap in different approaches.
5. REFERENCES


