Towards Unified Spoken Language Understanding Decoding via Label-aware Compact Linguistics Representations

Zhihong Zhu, Xuxin Cheng, Zhiqi Huang, Dongsheng Chen, Yuexian Zou

School of ECE, Peking University, China
{zhihongzhu, chengxx, chends}@stu.pku.edu.cn
{zhiqihuang, zouyx}@pku.edu.cn

Abstract

Joint intent detection and slot filling models have shown promising success in recent years due to the high correlations between the two tasks. However, previous works independently decode the two tasks, which could result in misaligned predictions for both tasks. To address this shortcoming, we propose a novel method named Label-aware Compact Linguistics Representation (LCLR), which leverages label embeddings to jointly guide the decoding process. Concretely, LCLR projects both task-specific hidden states into a joint label latent space, where both task-specific hidden states could be concisely represented as linear combinations of label embeddings. Such feature decomposition of task-specific hidden states increases the representing power for the linguistics of utterance. Extensive experiments on two single- and multi-intent SLU benchmarks prove that LCLR can learn more discriminative label information than previous separate decoders, and consistently outperform previous state-of-the-art methods across all metrics. More encouragingly, LCLR can be applied to boost the performance of existing approaches, making it easy to be incorporated into any existing SLU models.

1 Introduction

Spoken Language Understanding (SLU) plays a critical role in the task-oriented dialogue system (Tur and De Mori, 2011; Qin et al., 2021c). A typical SLU task mainly includes two subtasks, i.e., Intent Detection (ID) and Slot Filling (SF). Given by an utterance expressed in natural language from the user, ID aims to identify the intent of the user (e.g., GetWeather), and SF aims to fill the slot for each token in the utterance (e.g., location, time). Recent studies (Gangadharaiah and Narayanaswamy, 2019; Qin et al., 2020) find that users also express more than one intent in an utterance in many scenarios. Thus, multi-intent SLU is derived, attracting increasing attention.

Since the two tasks are highly related, a bunch of joint models (Huang et al., 2021; Qin et al., 2022; Chen et al., 2022a; Xing and Tsang, 2022a; Zhu et al., 2023) are proposed to tackle this two tasks jointly. Although achieving promising progress, the main technical challenges remain: Dislocation of the decoding process, where the updated decoding processes for the two tasks are completely isolated. This results in one type of information being unable to propagate to the other type of information in the updated decoding process, making it easier for the model’s predictions for the two tasks to become misaligned. In general, existing models solely employ the two tasks’ information with pipeline decoding method. This leaves us with a question: Can we simultaneously decode intent and slot labels in a unified decoding process to fully incorporate the dual-task correlative information?

Recent works have provided some first insights into jointly decoding the two tasks. Xu and Sarikaya (2013) extracted features through CNN
layers and model the dependencies between intent labels and slot tokens. Xing and Tsang (2022b) combined task-specific hidden states with label information using linear layers and dot products for enhancing decoding. However, their methods introduce additional parameters and still attempt to perform decoding in different task hidden spaces, which sever correlations between the two tasks.

To effectively and efficiently address the two tasks’ gap, we propose to learn a joint label latent space based on label embeddings to jointly guide the SLU decoding process. For this purpose, we propose a novel method named Label-aware Compact Linguistics Representation mechanism (LCLR), which uses the same parametric model to project and reformulate both task-specific hidden states. In detail, LCLR projects the task-specific hidden states into a joint latent space in best approximation algorithm (del Pino and Galaz, 1995), where the task-specific hidden states could be concisely represented as linear combinations of label embeddings. Such feature decomposition of task-specific hidden states increases representing power for the linguistics of utterance. In this manner, both intent-specific and slot-specific hidden states are represented with the distributions over the same sets of label hidden variables, which can be guided by the dual-task inter-dependencies conveyed in the learned label embeddings.

We conduct extensive experiments on both single-intent and multi-intent SLU benchmarks. The results show it can empower the different SLU models to consistently achieve better performance. Further analysis also demonstrates the advantages of our proposed LCLR.

Overall, our contributions are three-fold:

- We are the first to incorporate the label information into task-specific hidden states to jointly decode the SLU tasks from a linguistics representation perspective in a non-parametric manner.
- More encouragingly, LCLR is general and suitable for different SLU architectures.
- Comprehensive experiments on both single-/multi-intent SLU benchmarks demonstrate the effectiveness and superiority of LCLR.

2 Approach

2.1 Preliminaries

Single-intent SLU Given an input utterance \( x \), single intent detection and slot filling aims to output an intent label \( y^I \) and slots sequence \( y^S \) = \( (y_{1}^S, \ldots, y_{m}^S) \), where \( n \) denotes the length of \( x \).

Multi-intent SLU This means the SLU model should output an intent label set \( y^I = (y_{1}^I, \ldots, y_{m}^I) \) and slots sequence \( y^S \) = \( (y_{1}^S, \ldots, y_{m}^S) \), where \( m \) denotes the number of intents expressed in \( x \).

A generic SLU model Given an input utterance \( x = \{x_i\}_{1}^{n} \), the input hidden states \( h \) can be generated by utterance encoder, i.e., self-attentive encoder (Qin et al., 2020, 2021b), pre-trained model (Chen et al., 2022b; Cheng et al., 2023). Then \( h \) are fed to two different BiLSTMs (Hochreiter and Schmidhuber, 1997) to obtain intent-specific hidden states \( h^I \) and slot-specific hidden states \( h^S \) for intent detection and slot filling task, respectively. Eventually, a joint training scheme is adopted to optimize intent detection and slot filling simultaneously.

2.2 Label-aware Compact Linguistics Representations

Intent detection As for intent detection, instead of directly utilizing the intent-specific hidden states \( h^I \) to predict the intents labels, we first construct a joint label latent space \( \mathcal{T} \) with \( |I| + |S| \) label embeddings as basis \( \{v_{1}^I, \ldots, v_{|I|}^I, v_{1}^S, \ldots, v_{|S|}^S\} \). Then each intent-specific hidden token \( h_{i}^I \) is projected onto \( \mathcal{T} \) to obtain its linear approximation of a specific task \( \hat{h}_{i}^I = \sum_{j=1}^{|I|} w_{i,j}^I v_{j}^I \), where \( w_{i,j}^I \in \mathbb{R}^{|I|} \) could be computed as \( w_{i,j}^I = G_{i}^{-1}b_{i,j}^I \). The Gram matrix \( G_{i}^I \) and \( b_{i,j}^I \) can be formulated as follows:

\[
G_{i}^I = \begin{bmatrix}
\langle v_{1}^I, v_{1}^I \rangle & \cdots & \langle v_{|I|}^I, v_{|I|}^I \rangle \\
\vdots & \ddots & \vdots \\
\langle v_{1}^I, v_{|I|}^I \rangle & \cdots & \langle v_{|I|}^I, v_{|I|}^I \rangle
\end{bmatrix},
\]

\[
b_{i,j}^I = \begin{bmatrix}
\langle h_{i}^I, v_{1}^I \rangle \\
\vdots \\
\langle h_{i}^I, v_{|I|}^I \rangle
\end{bmatrix}.
\]

To note, we assume \( \{v_{1}^I, \ldots, v_{|I|}^I, v_{1}^S, \ldots, v_{|S|}^S\} \) are linearly independent, as each vector represents the concept of a label that should not be a linear combination of other label vectors. Therefore, \( G_{i}^I \) is guaranteed to be positive definite and have an inverse. After obtaining \( w_{i} \), these projection weights
can be viewed as scores of how likely this token of utterance \( x \) belongs to each intent \( y^I_i \). Then we treat it as a single-/multi-label classification task for single-/multi-intent SLU and generate the logits \( \hat{y}^I_i = \sigma(w^I_i) \) where \( \sigma \) denotes nonlinear function. The final output sentence-level intents are obtained via token-level intent voting over \( \hat{y}^I \).

**Slot filling** As for slot filling, the score \( w^S_i \) of each token in \( x \) can be derived like Eq. 1 and Eq. 2. Subsequently, we utilize a softmax classifier and an argmax function sequentially to generate the slot label distribution for each word:

\[
\hat{y}^S_i = \text{argmax}(\text{softmax}(w^S_i)),
\]

where \( \hat{y}^S_i \) is the predicted slot of the \( i \)-th token in the input utterance \( x \).

**Joint training** Owing to the strong correlation between intents and slots, joint models are utilized to consider the two tasks together and update parameters. The training objective of single-/multi-intent detection task is:

\[
\text{CE}(\hat{y}, y) = \hat{y} \log(y) + (1 - \hat{y}) \log(1 - y),
\]

\[
\mathcal{L}_{ID} = - \sum_{i=1}^{n} \sum_{j=1}^{N_i} \text{CE}(\hat{y}^I_i[j,I], y^I_i[j,I]),
\]

where \( N_i \) denotes the number of intent labels. Similarly, the training objective of slot filling task is defined as:

\[
\mathcal{L}_{SF} = - \sum_{i=1}^{n} \sum_{j=1}^{N_i} \hat{y}^S_i[j,S] \log(y^S_i[j,S]),
\]

Eventually, the total joint objective of LCLR is the weighted sum of two losses:

\[
\mathcal{L} = \alpha \cdot \mathcal{L}_{ID} + \beta \cdot \mathcal{L}_{SF},
\]

with two hyperparameters \( \alpha \) and \( \beta \) to balance.

### 3 Experiments

#### 3.1 Settings

**Datasets.** The statistics of datasets used in experiments are shown in Table 2.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>ATIS</th>
<th>SNIPS</th>
<th>MixATIS</th>
<th>MixSNIPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vocabulary Size</td>
<td>722</td>
<td>11241</td>
<td>766</td>
<td>11411</td>
</tr>
<tr>
<td>Avg. tokens per utterance</td>
<td>11.28</td>
<td>9.05</td>
<td>23.55</td>
<td>19.70</td>
</tr>
<tr>
<td>Intent categories</td>
<td>21</td>
<td>7</td>
<td>18</td>
<td>7</td>
</tr>
<tr>
<td>Slot categories</td>
<td>120</td>
<td>72</td>
<td>117</td>
<td>72</td>
</tr>
<tr>
<td>Training set size</td>
<td>4478</td>
<td>13084</td>
<td>13162</td>
<td>39776</td>
</tr>
<tr>
<td>Validation set size</td>
<td>893</td>
<td>700</td>
<td>828</td>
<td>2199</td>
</tr>
</tbody>
</table>

Table 2: Statistics of the benchmarks in single-/multi-intent SLU.

- Single-intent SLU: **SNIPS** (Coucke et al., 2018) has 13,084 utterances for training, 700 for validation, and 700 for testing.
### Table 3: Ablation study of our approach, which includes the state-of-the-art baseline models on single-intent and multi-intent benchmarks, respectively. Higher is better in all columns. ICLR and SCLR denote the intent-aware and slot-aware compact linguistics representations. Full Model represents the baseline model with LCLR.

<table>
<thead>
<tr>
<th>Single-intent SLU Methods</th>
<th>LCLR</th>
<th>Dataset: ATIS (Hemphill et al., 1990)</th>
<th>Dataset: SNIPS (Coucke et al., 2018)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ICLR</td>
<td>Slot (F1)</td>
<td>Intent (Acc)</td>
</tr>
<tr>
<td>HAN (a)</td>
<td>✓</td>
<td>97.2</td>
<td>99.1</td>
</tr>
<tr>
<td>HAN (b)</td>
<td>✓</td>
<td>97.5</td>
<td>99.2</td>
</tr>
<tr>
<td>Full Model</td>
<td>✓</td>
<td>97.6</td>
<td>99.4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Multi-intent SLU Methods</th>
<th>LCLR</th>
<th>Dataset: MixATIS (Qin et al., 2020)</th>
<th>Dataset: MixSNIPS (Qin et al., 2020)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ICLR</td>
<td>Slot (F1)</td>
<td>Intent (Acc)</td>
</tr>
<tr>
<td>Co-guiding Net (a)</td>
<td>✓</td>
<td>89.8</td>
<td>79.1</td>
</tr>
<tr>
<td>Co-guiding Net (b)</td>
<td>✓</td>
<td>90.2</td>
<td>79.2</td>
</tr>
<tr>
<td>Full Model</td>
<td>✓</td>
<td>90.2</td>
<td>79.4</td>
</tr>
</tbody>
</table>

### Evaluation metrics

In our experiments, we evaluate the performance of models on the widely-used spoken language understanding metrics (Goo et al., 2018), i.e., accuracy (Acc) for intent-detection, F1 score for slot filling, and overall accuracy for the sentence-level semantic frame parsing. In particular, overall accuracy denotes the ratio of utterances whose intents and slots are all correctly predicted.

### 3.2 Baselines

In our experiments, we choose seven SLU models including both single-intent and multi-intent SLU with different structures as baseline models, i.e., 1) JointBERT (Chen et al., 2019), 2) LR-Transformer (Cheng et al., 2021), 3) Co-Interactive (Qin et al., 2021a), 4) HAN (Chen et al., 2022a), 5) GL-GIN (Qin et al., 2021b), 6) Song et al. (Song et al., 2022), and 7) Co-guiding Net (Xing and Tsang, 2022a). In detail, to demonstrate the effectiveness of LCLR, we compare the performance of these models with and without LCLR.

### 3.3 Results

#### Main results

The experimental results of different categories of SLU models on corresponding benchmark datasets are reported in Table 1. As shown, our proposed LCLR can consistently boost all baselines across all metrics, where the HAN and Co-guiding Net with LCLR achieves the greatest improvements, respectively. It is noteworthy that the multi-intent SLU models with LCLR result in a more significant increase in performance compared to single-intent SLU ones with LCLR. We attribute this to LCLR can decouple utterances into linear representations of label information, enhancing the linguistic features of the utterances and facilitating the discriminatory power for different labels.

#### Ablation study

We select two mainstream SLU models, i.e., HAN and Co-guiding Net, to evaluate the contribution of each proposed module, i.e., intent-aware compact linguistics and slot-aware compact linguistics representations (cf. Table 3). As we can see, each component in our proposed approach can boost the performances of baselines over all metrics, verifying the effectiveness of our approach.

- Effect of ICLR/SCLR. Setting (a)/(b) in Table 3 shows that ICLR/SCLR can successfully boost baselines, demonstrating how ICLR/SCLR exploits the different task-specific label information to jointly guide the decoding process.

- Effect of LCLR. Since the ICLR and LCLR can improve the performance from different information sources, combining them can lead
to the most prominent improvement across all metrics (see Full Model), with up to 92.4% and 89.5% overall acc for ATIS and SNIPS in terms of HAN; 52.0% and 78.1% overall acc for MixATIS and MixSNIPS in terms of Co-guiding Net, respectively.

**4 Conclusion**

We propose a novel method called Label-aware Compact Linguistics Representation (LCLR) to jointly guide the decoding process. In the joint label latent space, both task-specific hidden states are concisely represented as the linear combinations of label embeddings, enhancing representing power for the linguistics of utterance. This approach allows the decoding process to be guided by the dual-task inter-dependencies conveyed in the learned label embeddings. Experimental results on both single- and multi-intent SLU benchmarks demonstrate LCLR can consistently empower various SLU models to achieve better performance.

**Limitations**

Although LCLR shows great potential for unifying the SLU decoding process, existing SLU models experiment on a set of predefined labels (closed domain), and our LCLR can handle the case of missing predefined labels in the train. It is interesting to try to perform LCLR on a more challenging task of detecting out-of-domain (OOD) detection where unseen intents/slots are not available.

**Acknowledgements**

We thank all anonymous reviewers for their constructive comments. This paper was partially supported by Shenzhen Science & Technology Research Program (No:GXWD20201231165807007-20200814115301001) and NSFC (No: 62176008).

**References**

Dongsheng Chen, Zhiqi Huang, Xian Wu, Shen Ge, and Yuexian Zou. 2022a. Towards joint intent detection and slot filling via higher-order attention. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022.*


Rashmi Gangadharaiah and Balakrishnan Narayanawamy. 2019. Joint multiple intent


A Appendix

A.1 Best Approximation in a Hilbert Space

**Theorem** Let $S$ be a Hilbert space with inner product $\langle \cdot, \cdot \rangle$ and induced norm $\| \cdot \|$, and let $T$ be a finite dimensional subspace \(^1\). Given an arbitrary $x \in S$, there is exactly one $\hat{x} \in T$ such that

$$x - \hat{x} \perp T,$$

(8)

meaning $\langle x - \hat{x}, y \rangle = 0$ for all $y \in T$, and this $\hat{x}$ is the closet point in $T$ to $x$; that is, $\hat{x}$ is the unique minimizer to

$$\min_{y \in T} \| x - y \|.$$

(9)

**Proof** Let $\hat{x}$ be the vector which obeys $\hat{e} = x - \hat{x} \perp T$. Let $y$ be any other vector in $T$, and set $e = x - y$. Note that

$$\| e \|^2 = \| x - y \|^2 = \| \hat{e} - (y - \hat{x}) \|^2$$

$$= \| \hat{e} - (y - \hat{x}) \|^2 + \| y - \hat{x} \|^2 - \langle \hat{e}, y - \hat{x} \rangle - \langle y - \hat{x}, \hat{e} \rangle$$

(10)

Since $y - \hat{x} \in T$ and $\hat{e} \perp T$, $\langle \hat{e}, y - \hat{x} \rangle = 0$,

$$\langle \hat{e}, y - \hat{x} \rangle = \langle y - \hat{x}, \hat{e} \rangle = 0,$$

(11)

and so

$$\| e \|^2 = \| \hat{e} \|^2 + \| y - \hat{x} \|^2.$$

Thus all three quantities in the expression above are positive and $\| y - \hat{x} \| > 0$, $\| e \| > \| \hat{e} \|$.\(^{12}\)

**Computing the best approximation** Let $N$ be the dimension of $T$, and let $v_1, \ldots, v_N$ be a basis for $T$. We can find coefficients $a_1, \ldots, a_N$ such that

$$\hat{x} = a_1 v_1 + a_2 v_2 + \cdots + a_N v_N.$$ \(^{13}\)

(13)

According to the orthogonality principle, the $a_n$ must obey

$$\langle x, v_n \rangle = \sum_{n,k=1}^N a_k \langle v_k, v_n \rangle.$$ \(^{14}\)

(14)

We are left with a set of $N$ linear equations with $N$ unknowns:

$$
\begin{bmatrix}
\langle v_1, v_1 \rangle & \langle v_2, v_1 \rangle & \cdots & \langle v_N, v_1 \rangle \\
\langle v_1, v_2 \rangle & \langle v_2, v_2 \rangle & \cdots & \langle v_N, v_2 \rangle \\
\vdots & \vdots & \ddots & \vdots \\
\langle v_1, v_N \rangle & \cdots & \langle v_N, v_N \rangle \\
\end{bmatrix}
\begin{bmatrix}
a_1 \\
a_2 \\
\vdots \\
a_N \\
\end{bmatrix}
= \begin{bmatrix}
\langle x, v_1 \rangle \\
\langle x, v_2 \rangle \\
\vdots \\
\langle x, v_N \rangle \\
\end{bmatrix}.$$ \(^{15}\)

(15)

\(^1\)The same results hold when $T$ is infinite dimensional and is closed.

\(^{12}\)https://github.com/pytorch/pytorch/

\(^{13}\)With the work above, this means that a necessary and sufficient condition for $\langle x - \hat{x}, y \rangle = 0$ for all $y \in T$ is to have

$$\hat{x} = \sum_{n=1}^N a_n v_n,$$

(16)

where $a$ satisfies $Ga = b$; where $b_n = \langle x, v_n \rangle$ and $G_{n,k} = \langle v_n, v_k \rangle$.

Since $G$ is square and invertible, there is exactly one such $a$, and hence exactly one $\hat{x}$ that obeys the condition

$$x - \hat{x} \perp T.$$ \(^{16}\)

(17)

A.2 Implementation Details

We implemented all the models used in our experiments using PyTorch (Paszke et al., 2019) (ver. 1.10.1)\(^2\) one 1 Nvidia V100. We run the baselines also on the same computing environment, using the configuration file they provided.
ACL 2023 Responsible NLP Checklist

A For every submission:

✔️ A1. Did you describe the limitations of your work?
   \[\text{In Section Limitations}\]

☒ A2. Did you discuss any potential risks of your work?
   \[\text{This paper does not involve any data collection and release thus there are no privacy issues. All the datasets used in this paper are publicly available and widely adopted by researchers to test the performance of SLU models.}\]

✔️ A3. Do the abstract and introduction summarize the paper’s main claims?
   \[\text{In Section Abstract and Section 1. Introduction}\]

☒ A4. Have you used AI writing assistants when working on this paper?
   \[\text{Left blank}\]

B ✔ Did you use or create scientific artifacts?
   \[\text{In section 3. Experiments}\]

✔️ B1. Did you cite the creators of artifacts you used?
   \[\text{In section 3. Experiments}\]

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
   \[\text{Not applicable. Left blank}\]

✔️ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
   \[\text{In section 3. Experiments}\]

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
   \[\text{Not applicable. Left blank}\]

✔️ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
   \[\text{In section 3. Experiments}\]

✔️ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
   \[\text{In section 3. Experiments}\]

C ✔ Did you run computational experiments?
   \[\text{In section 3. Experiments}\]

✔️ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
   \[\text{In section 3. Experiments and section Appendix A.2. Implementation Details}\]

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.
C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?

We run the baselines on the same computing environment, using the configuration file they provided.

C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?

In section 3. Experiments.

C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?

In section 3. Experiments.

D Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?

No response.

D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants’ demographic (e.g., country of residence)?

No response.

D3. Did you discuss whether and how consent was obtained from people whose data you’re using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?

No response.

D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?

No response.

D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?

No response.