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 **L**abel-aware **C**ompact **L**inguistics **R**epresentation (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.

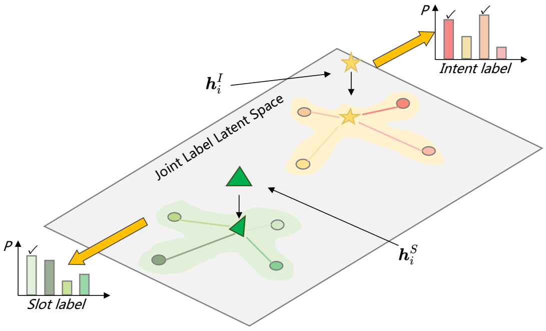


Figure 1: Illustration of our proposed LCLR. Blue/green circles denote slot labels and red/yellow circles denote intent labels. The different colored arrows indicate the coefficient by which the hidden state projection is linearly decomposed for this label. The shorter the distance, the greater the probability that the label is correct. ✓denotes that the label is selected as the prediction.

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

[*] Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2023, pages 12523–12531
July 9-14, 2023 ©2023 Association for Computational Linguistics

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 severs 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 **L**abel-aware **C**ompact **L**inguistics **R**epresentation 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 label 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 $\boldsymbol{x}$, single intent detection and slot filling aims to output an intent label y^I and slots sequence $\boldsymbol{y}^S = (y_1^S, \ldots, y_n^S)$, where n denotes the length of $\boldsymbol{x}$.

Multi-intent SLU This means the SLU model should output an intent label set $\boldsymbol{y}^I = (y_1^I, \ldots, y_m^I)$ and slots sequence $\boldsymbol{y}^S = (y_1^S, \ldots, y_n^S)$, where m denotes the number of intents expressed in $\boldsymbol{x}$.

A generic SLU model Given an input utterance $\boldsymbol{x} = \{x_i\}_1^n$, the input hidden states $\boldsymbol{h}$ 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 $\boldsymbol{h}$ are fed to two different BiLSTMs (Hochreiter and Schmidhuber, 1997) to obtain intent-specific hidden states $\boldsymbol{h}^I$ and slot-specific hidden states $\boldsymbol{h}^S$ 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 $\boldsymbol{h}^I$ to predict the intents labels, we first construct a joint label latent space $\mathcal{T}$ with $|I|+|S|$ label embeddings as basis $\{\boldsymbol{v}_1^I, \ldots, \boldsymbol{v}_{|I|}^I, \boldsymbol{v}_1^S, \ldots, \boldsymbol{v}_{|S|}^S\}$. Then each intent-specific hidden token $\boldsymbol{h}_i^I$ is projected onto $\mathcal{T}$ to obtain its linear approximation of a specific task $\hat{\boldsymbol{h}}_i^I = \sum_{j=1}^{|I|} w_{[i,j]}^I \boldsymbol{v}_j^I$, where $\boldsymbol{w}_i^I \in \mathbb{R}^{|I|}$ could be computed as $\boldsymbol{w}_i^I = \boldsymbol{G}_i^{I^{-1}} \boldsymbol{b}_i^I$. The Gram matrix $\boldsymbol{G}_i^I$ and $\boldsymbol{b}_i^I$ can be formulated as follows:

$$\boldsymbol{G}_i^I = \begin{bmatrix} \langle \boldsymbol{v}_1^I, \boldsymbol{v}_1^I \rangle & \cdots & \left\langle \boldsymbol{v}_{|I|}^I, \boldsymbol{v}_1^I \right\rangle \\ \vdots & \ddots & \vdots \\ \left\langle \boldsymbol{v}_1^I, \boldsymbol{v}_{|I|}^I \right\rangle & \cdots & \left\langle \boldsymbol{v}_{|I|}^I, \boldsymbol{v}_{|I|}^I \right\rangle \end{bmatrix}, \tag{1}$$

$$\boldsymbol{b}_i^I = \begin{bmatrix} \langle \boldsymbol{h}_i^I, \boldsymbol{v}_1^I \rangle \\ \vdots \\ \left\langle \boldsymbol{h}_i^I, \boldsymbol{v}_{|I|}^I \right\rangle \end{bmatrix}. \tag{2}$$

To note, we assume $\{\boldsymbol{v}_1^I, \ldots, \boldsymbol{v}_{|I|}^I, \boldsymbol{v}_1^S, \ldots, \boldsymbol{v}_{|S|}^S\}$ are linearly independent, as each vector represents the concept of a label that should not be a linear combination of other label vectors. Therefore, $\boldsymbol{G}_i^I$ is guaranteed to be positive definite and have an inverse. After obtaining $\boldsymbol{w}_i$, these projection weights

Single-intent SLU Methods	Dataset: ATIS (Hemphill et al., 1990)			Dataset: SNIPS (Coucke et al., 2018)		
	Slot (F1)	Intent (Acc)	Overall (Acc)	Slot (F1)	Intent (Acc)	Overall (Acc)
JointBERT (Chen et al., 2019)	96.1	97.5	88.2	97.0	98.6	92.8
with LCLR	**96.6**	**97.8**	**88.8**	**97.3**	**98.9**	**93.0**
LR-Transformer (Cheng et al., 2021)	96.1	98.2	87.2	94.8	98.4	88.4
with LCLR	**96.7**	**98.5**	**87.8**	**95.2**	**98.7**	**88.9**
Co-Interactive (Qin et al., 2021a)	95.9	98.8	90.3	95.9	97.7	87.4
with LCLR	**96.3**	**99.0**	**91.2**	**96.3**	**98.1**	**88.0**
HAN (Chen et al., 2022a)	97.2	99.1	91.8	96.5	98.5	88.7
with LCLR	**97.6**	**99.4**	**92.4**	**96.8**	**98.9**	**89.5**
Multi-intent SLU Methods	Dataset: MixATIS (Qin et al., 2020)			Dataset: MixSNIPS (Qin et al., 2020)		
	Slot (F1)	Intent (Acc)	Overall (Acc)	Slot (F1)	Intent (Acc)	Overall (Acc)
GL-GIN (Qin et al., 2021b)	88.3	76.3	43.5	94.9	95.6	75.4
with LCLR	**88.6**	**77.1**	**44.8**	**95.3**	**96.1**	**75.8**
Song et al. (Song et al., 2022)	88.5	75.0	48.2	95.0	95.5	75.9
with LCLR	**88.9**	**75.6**	**49.3**	**95.4**	**95.9**	**76.5**
Co-guiding Net (Xing and Tsang, 2022a)	89.8	79.1	51.3	95.1	97.7	77.5
with LCLR	**90.2**	**79.4**	**52.0**	**95.5**	**98.1**	**78.1**

Table 1: Performance on two benchmark datasets. Higher is better in all columns. We conducted 5 runs with different seeds for all experiments, the t-tests indicate that $p < 0.01$. As we can see, all the baseline models with significantly different structures enjoy a comfortable improvement with our LCLR.

can be viewed as scores of how likely this token of utterance $\boldsymbol{x}$ 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 $\hat{y}_i^I = \sigma(\boldsymbol{w}_i^I)$ where σ denotes nonlinear function. The final output sentence-level intents are obtained via token-level intent voting over $\hat{\boldsymbol{y}}^I$.

Slot filling As for slot filling, the score $\boldsymbol{w}_i^S$ of each token in $\boldsymbol{x}$ can be derived like Eq. 1 and Eq. 2. Subsequently, we utilize a $\mathrm{softmax}$ classifier and an argmax function sequentially to generate the slot label distribution for each word:

$$\hat{y}_i^S = \mathrm{argmax}(\mathrm{softmax}(\boldsymbol{w}_i^S)), \tag{3}$$

where $\hat{y}_i^S$ is the predicted slot of the i-th token in the input utterance $\boldsymbol{x}$.

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:

$$\mathrm{CE}(\hat{y}, y) = \hat{y}\log(y) + (1-\hat{y})\log(1-y), \tag{4}$$

$$\mathcal{L}_{ID} = -\sum_{i=1}^{n}\sum_{j=1}^{N_I}\mathrm{CE}(\hat{y}_i^{[j,I]}, y_i^{[j,I]}), \tag{5}$$

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

$$\mathcal{L}_{SF} = -\sum_{i=1}^{n}\sum_{j=1}^{N_S}\hat{y}_i^{[j,S]}\log(y_i^{[j,S]}), \tag{6}$$

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

$$\mathcal{L} = \alpha \cdot \mathcal{L}_{ID} + \beta \cdot \mathcal{L}_{SF}, \tag{7}$$

with two hyperparameters α and β to balance.

3 Experiments

3.1 Settings

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

Dataset	ATIS	SNIPS	MixATIS	MixSNIPS
Vocabulary Size	722	11241	766	11411
Avg. tokens per utterance	11.28	9.05	23.55	19.70
Intent categories	21	7	18	7
Slot categories	120	72	117	72
Training set size	4478	13084	13162	39776
Validation set size	500	700	759	2198
Test set size	893	700	828	2199

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.

Single-intent SLU Methods	LCLR		Dataset: ATIS (Hemphill et al., 1990)			Dataset: SNIPS (Coucke et al., 2018)		
	ICLR	SCLR	Slot (F1)	Intent (Acc)	Overall (Acc)	Slot (F1)	Intent (Acc)	Overall (Acc)
HAN			97.2	99.1	91.8	96.5	98.5	88.7
(a)	✓		97.3	99.3	92.0	96.5	98.7	89.0
(b)		✓	97.5	99.2	92.1	96.7	98.6	89.2
Full Model	✓	✓	**97.6**	**99.4**	**92.4**	**96.8**	**98.9**	**89.5**

Multi-intent SLU Methods	LCLR		Dataset: MixATIS (Qin et al., 2020)			Dataset: MixSNIPS (Qin et al., 2020)		
	ICLR	SCLR	Slot (F1)	Intent (Acc)	Overall (Acc)	Slot (F1)	Intent (Acc)	Overall (Acc)
Co-guiding Net			89.8	79.1	51.3	95.1	97.7	77.5
(a)	✓		90.0	79.3	51.6	95.2	97.9	77.8
(b)		✓	90.2	79.2	51.7	95.4	97.8	77.7
Full Model	✓	✓	**90.2**	**79.4**	**52.0**	**95.5**	**98.1**	**78.1**

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.

ATIS (Hemphill et al., 1990) has 4,478 utterances for training, 500 for validation, and 893 for testing.

- Multi-intent SLU: **MixSNIPS** (Qin et al., 2020) is constructed from **SNIPS** which comprises 39,776/2,198/2,199 utterances for training, validation and testing, separately. **MixATIS** (Qin et al., 2020) is collected from **ATIS**, which contains 13,161/759/828 utterances for training, validation and testing, respectively.

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.

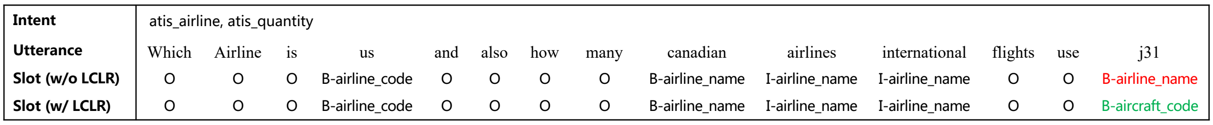
3.3 Results

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



Intent	atis_airline, atis_quantity															
Utterance	Which	Airline	is	us	and	also	how	many	canadian	airlines	international	flights	use	j31
Slot (w/o LCLR)	O	O	O	B-airline_code	O	O	O	O	B-airline_name	I-airline_name	I-airline_name	O	O	B-airline_name
Slot (w/ LCLR)	O	O	O	B-airline_code	O	O	O	O	B-airline_name	I-airline_name	I-airline_name	O	O	B-aircraft_code

Figure 2: Case study between **Co-guiding Net** with and without LCLR on the **MixATIS** dataset. The green slot is correct while the red one is wrong. Better viewed in color.

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.

Qualitative analysis We conduct a qualitative analysis to understand our approach more thoroughly. As shown in Figure 2, we can see that **Co-guiding Net** with LCLR predicts the slot label "`B-aircraft_code`" of token "`j31`" correctly, while **Co-guiding Net** without LCLR predicts it as "`O`" incorrectly. This also demonstrates that our proposed LCLR can fully learn the distinguishing information of different labels during the decoding process, boosting SLU performance.

4 Conclusion

We propose a novel method called **L**abel-aware **C**ompact **L**inguistics **R**epresentation (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*.

Lisung Chen, Nuo Chen, Yuexian Zou, Yong Wang, and Xinzhong Sun. 2022b. A transformer-based threshold-free framework for multi-intent NLU. In *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022*.

Qian Chen, Zhu Zhuo, and Wen Wang. 2019. BERT for joint intent classification and slot filling. *CoRR*, abs/1902.10909.

Lizhi Cheng, Weijia Jia, and Wenmian Yang. 2021. An effective non-autoregressive model for spoken language understanding. In *the 30th ACM International Conference on Information and Knowledge Management, CIKM 2021*.

Xuxin Cheng, Bowen Cao, Qichen Ye, Zhihong Zhu, Hongxiang Li, and Yuexian Zou. 2023. Ml-lmcl: Mutual learning and large-margin contrastive learning for improving asr robustness in spoken language understanding. In *Findings of the Association for Computational Linguistics: ACL 2023*.

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, Maël Primet, and Joseph Dureau. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *CoRR*, abs/1805.10190.

Guido E del Pino and Hector Galaz. 1995. Statistical applications of the inverse gram matrix: A revisitation. *Brazilian Journal of Probability and Statistics*, pages 177–196.

Rashmi Gangadharaiah and Balakrishnan Narayanaswamy. 2019. Joint multiple intent detection and slot labeling for goal-oriented dialog. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019*.

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 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018*.

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

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Zhiqi Huang, Fenglin Liu, Peilin Zhou, and Yuexian Zou. 2021. Sentiment injected iteratively co-interactive network for spoken language understanding. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2021*.

Fenglin Liu, Yuanxin Liu, Xuancheng Ren, Xiaodong He, and Xu Sun. 2019. Aligning visual regions and textual concepts for semantic-grounded image representations. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019*.

Fenglin Liu, Xian Wu, Shen Ge, Wei Fan, and Yuexian Zou. 202. Federated learning for vision-and-language grounding problems. In *the Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019*.

Libo Qin, Qiguang Chen, Tianbao Xie, Qixin Li, Jian-Guang Lou, Wanxiang Che, and Min-Yen Kan. 2022. GL-CLeF: A global-local contrastive learning framework for cross-lingual spoken language understanding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, ACL 2022*.

Libo Qin, Tailu Liu, Wanxiang Che, Bingbing Kang, Sendong Zhao, and Ting Liu. 2021a. A co-interactive transformer for joint slot filling and intent detection. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2021*.

Libo Qin, Fuxuan Wei, Tianbao Xie, Xiao Xu, Wanxiang Che, and Ting Liu. 2021b. GL-GIN: fast and accurate non-autoregressive model for joint multiple intent detection and slot filling. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021*.

Libo Qin, Tianbao Xie, Wanxiang Che, and Ting Liu. 2021c. A survey on spoken language understanding: Recent advances and new frontiers. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021*.

Libo Qin, Xiao Xu, Wanxiang Che, and Ting Liu. 2020. Towards fine-grained transfer: An adaptive graph-interactive framework for joint multiple intent detection and slot filling. In *Findings of the Association for Computational Linguistics: EMNLP 2020*.

Mengxiao Song, Bowen Yu, Quangang Li, Yubin Wang, Tingwen Liu, and Hongbo Xu. 2022. Enhancing joint multiple intent detection and slot filling with global intent-slot co-occurrence. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022*.

Gokhan Tur and Renato De Mori. 2011. *Spoken language understanding: Systems for extracting semantic information from speech*. John Wiley & Sons.

Bowen Xing and Ivor W. Tsang. 2022a. Co-guiding net: Achieving mutual guidances between multiple intent detection and slot filling via heterogeneous semantics-label graphs. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022*.

Bowen Xing and Ivor W. Tsang. 2022b. Group is better than individual: Exploiting label topologies and label relations for joint multiple intent detection and slot filling. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022*.

Puyang Xu and Ruhi Sarikaya. 2013. Convolutional neural network based triangular CRF for joint intent detection and slot filling. In *2013 IEEE Workshop on Automatic Speech Recognition and Understanding, Olomouc, Czech Republic, December 8-12, 2013*.

Zhihong Zhu, Weiyuan Xu, Xuxin Cheng, Tengtao Song, and Yuexian Zou. 2023. A dynamic graph interactive framework with label-semantic injection for spoken language understanding. In *2023 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2023*.

A Appendix

A.1 Best Approximation in a Hibert Space

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

$$\boldsymbol{x} - \hat{\boldsymbol{x}} \perp \mathcal{T}, \tag{8}$$

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

$$\underset{\boldsymbol{y}\in\mathcal{T}}{\text{minimize}} \|\boldsymbol{x} - \boldsymbol{y}\|. \tag{9}$$

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

$$\begin{aligned} \|\boldsymbol{e}\|^2 &= \|\boldsymbol{x} - \boldsymbol{y}\|^2 = \|\hat{e} - (\boldsymbol{y} - \hat{\boldsymbol{x}})\|^2 \\ &= \langle \hat{e} - (\boldsymbol{y} - \hat{\boldsymbol{x}}), \hat{e} - (\boldsymbol{y} - \hat{\boldsymbol{x}}) \rangle \qquad . \\ &= \|\hat{e}\|^2 + \|\boldsymbol{y} - \hat{\boldsymbol{x}}\|^2 - \langle \hat{e}, \boldsymbol{y} - \hat{\boldsymbol{x}} \rangle - \langle \boldsymbol{y} - \hat{\boldsymbol{x}}, \hat{e} \rangle \end{aligned} \tag{10}$$

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

$$\langle \hat{e}, \boldsymbol{y} - \hat{\boldsymbol{x}} \rangle = \langle \boldsymbol{y} - \hat{\boldsymbol{x}}, \hat{e} \rangle = 0, \tag{11}$$

and so

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

Thus all three quantities in the expression above are positive and $\|\boldsymbol{y} - \hat{\boldsymbol{x}}\| > 0$,

$$\|\boldsymbol{e}\| > \|\hat{\boldsymbol{e}}\|. \tag{12}$$

Computing the best approximation Let N be the dimension of $\mathcal{T}$, and let $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n$ be a basis for $\mathcal{T}$. We can find coefficients $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n$ such that

$$\hat{\boldsymbol{x}} = a_1\boldsymbol{v}_1 + a_2\boldsymbol{v}_2 + \cdots + a_N\boldsymbol{v}_N. \tag{13}$$

According to the orthogonality principle, the a_n must obey

$$\langle \boldsymbol{x}, \boldsymbol{v}_n \rangle = \sum_{n,k=1}^{N} a_k \langle \boldsymbol{v}_k, \boldsymbol{v}_n \rangle. \tag{14}$$

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

$$\begin{bmatrix} \langle \boldsymbol{v}_1, \boldsymbol{v}_1 \rangle & \langle \boldsymbol{v}_2, \boldsymbol{v}_1 \rangle & \cdots & \langle \boldsymbol{v}_N, \boldsymbol{v}_1 \rangle \\ \langle \boldsymbol{v}_1, \boldsymbol{v}_2 \rangle & \langle \boldsymbol{v}_2, \boldsymbol{v}_2 \rangle & & \langle \boldsymbol{v}_N, \boldsymbol{v}_2 \rangle \\ \vdots & & \ddots & \vdots \\ \langle \boldsymbol{v}_1, \boldsymbol{v}_N \rangle & \cdots & & \langle \boldsymbol{v}_N, \boldsymbol{v}_N \rangle \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_N \end{bmatrix} = \begin{bmatrix} \langle \boldsymbol{x}, \boldsymbol{v}_1 \rangle \\ \langle \boldsymbol{x}, \boldsymbol{v}_2 \rangle \\ \vdots \\ \langle \boldsymbol{x}, \boldsymbol{v}_N \rangle \end{bmatrix}. \tag{15}$$

The matrix on the left hand side above is called the Gram matrix $\boldsymbol{G}$ of the basis $\{\boldsymbol{v}_n\}$.

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

$$\hat{\boldsymbol{x}} = \sum_{n=1}^{N} a_n \boldsymbol{v}_n, \tag{16}$$

where $\boldsymbol{a}$ satisfies $\boldsymbol{G}\boldsymbol{a} = \boldsymbol{b}$; where $b_n = \langle \boldsymbol{x}, \boldsymbol{v}_n \rangle$ and $G_{k,n} = \langle \boldsymbol{v}_n, \boldsymbol{v}_k \rangle$.

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

$$\boldsymbol{x} - \hat{\boldsymbol{x}} \perp \mathcal{T}. \tag{17}$$

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.

[1] The same results hold when $\mathcal{T}$ is infinite dimensional and is closed.

[2] https://github.com/pytorch/pytorch/

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
In Section Limitations

☒ A2. Did you discuss any potential risks of your work?
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?
In Section Abstract and Section 1. Introduction.

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

In section 3. Experiments.

☑ B1. Did you cite the creators of artifacts you used?
In section 3. Experiments.

□ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
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)?
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?
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.?
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.
In section 3. Experiments.

C ☑ Did you run computational experiments?

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?
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.