CLID: A Chunk-Level Intent Detection Framework for Multiple Intent Spoken Language Understanding

Haojing Huang, Peijie Huang, *Member, IEEE*, Zhanbiao Zhu, Jia Li, and Piyuan Lin

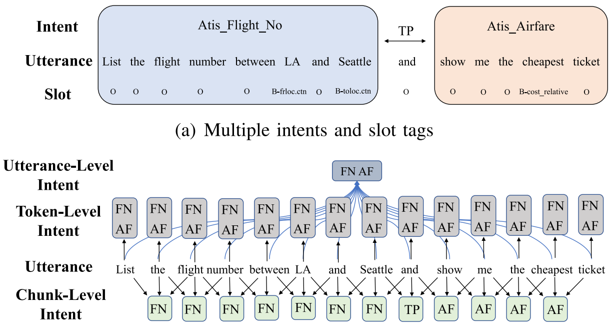
***Abstract*—Multi-intent spoken language understanding (SLU) that can handle an utterance containing multiple intents is more practical and attracts increasing attention. However, existing state-of-the-art models are either too coarse-grained (Utterance-level) or too fine-grained (Token-level) in intent detection, and thus may fail to recognize the intent transition point and the correct intents in an utterance. In this paper, we propose a Chunk-Level Intent Detection (CLID) framework, where we introduce a sliding window-based self-attention (SWSA) scheme for regional chunk intent detection. Based on the SWSA, an auxiliary task is introduced to identify the intent transition point in an utterance and obtain sub-utterances with a single intent. The intent of each sub-utterance is then predicted by assembling the intent predictions of the chunks (in a sliding window manner) within it. We conduct experiments on two public datasets, MixATIS and MixSNIPS, and the results show that our model achieves state-of-the-art performance.**

***Index Terms*—Chunk-level, intent detection, multiple intents, spoken language understanding.**

I. Introduction

Spoken language understanding (SLU) is a critical component of task-oriented dialogue systems, which aims to create a semantic frame that succinctly summarizes the user's request. Such a semantic frame is typically constructed using intent detection to identify users' intents and slot filling to extract relevant semantic constituents [1], [2], [3]. Since the two sub-tasks of intent detection and slot filling have a strong correlation, state-of-the-art models [4], [5], [6], [7], [8], [9], [10], [11], [12], [13] adopt joint models for modeling the relation between them.

In real-world scenarios, a user's utterance usually contains multiple intents. For example, a user who wants to inquire about the flight is also likely to pay attention to the ticket price. Fig. 1(a) shows an example with intents `Flight_no` and `Airfare`. To this end, a multi-intent SLU model that can handle one utterance containing multiple intents is shown to be more practical and attracts increasing attention [11], [12], [13], [14], [15].


Intent
Utterance
Slot
Atis_Flight_No
TP
Atis_Airfare
List the flight number between LA and Seattle
and
show me the cheapest ticket
O O O O O B-frloc.ctn O B-toloc.ctn
O
O O O B-cost_relative O
Utterance-Level Intent
FN AF
Token-Level Intent
FN AF
Utterance
List the flight number between LA and Seattle and show me the cheapest ticket
Chunk-Level Intent
FN FN FN FN FN FN FN TP AF AF AF AF

(a) Multiple intents and slot tags

(b) Prior intent detection frameworks vs. our chunk-level framework

Fig. 1. An example with multiple intents. "FN," "AF" and "TP" denote "Flight_No," "Airfare" and "Transition point".

Gangadharaiah and Narayanaswamy [11] first proposed a joint model for multiple intent detection and slot filling. Their work introduced an utterance-level intent detection framework by using an intent context vector for predicting multiple intents. Qin et al. [12] also used utterance-level intent detection. In their work, an adaptive graph interaction framework (AGIF) was proposed to achieve fine-grained multi-intent integration for token-level slot filling. Qin et al. [13] proposed a non-autoregressive framework to accelerate the inference speed while achieving high accuracy. They used token-level multi-intent detection, which predicts multiple intents on each token.

Though achieving the promising performance, the intent detection of the existing multi-intent SLU joint models is either too coarse-grained (Utterance-level) or too fine-grained (Token-level) to capture the intent semantics, leading to two issues in multi-intent scenarios:

- They do not have an explicit model to learn and identify the intent transition points in multi-intent utterances. Identifying such transition points will help to further correctly recognize the intents of the sub-utterances. However, utterance-level intent detection methods ignore such intent transition points. While using the token-level mechanism, it is difficult to identify the intent transition point. For example, in the two-intents example shown in Fig. 1(a), the intent transition point cannot be determined simply by a token such as "and" (notice that "and" also appear in the first part of this example).
- Utterance-level intent detection models achieve satisfying performance in the single-intent scenario, but it is hard for them to compress all the necessary information of a multi-intent utterance into a fixed-length utterance context for intent detection. While token-level intent detection models predict intents on every token, which may lose

Manuscript received 5 May 2022; revised 3 September 2022; accepted 18 September 2022. Date of publication 30 September 2022; date of current version 25 October 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 71472068, in part by the Natural Science Foundation of Guangdong Province under Grant 2021A1515011864, and in part by the Special Funds for the Cultivation of Guangdong College Students through Scientific and Technological Innovation under Grant pdjh2022a007. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Sandro Cumani. *(Corresponding author: Peijie Huang.)*

The authors are with the College of Mathematics and Informatics, South China Agricultural University, Guangzhou 510642, China (e-mail: hjhuang@stu.scau.edu.cn; pjhuang@scau.edu.cn; zbzhu@stu.scau.edu.cn; jlee@stu.scau.edu.cn; pyuanlin@scau.edu.cn).

Digital Object Identifier 10.1109/LSP.2022.3211156

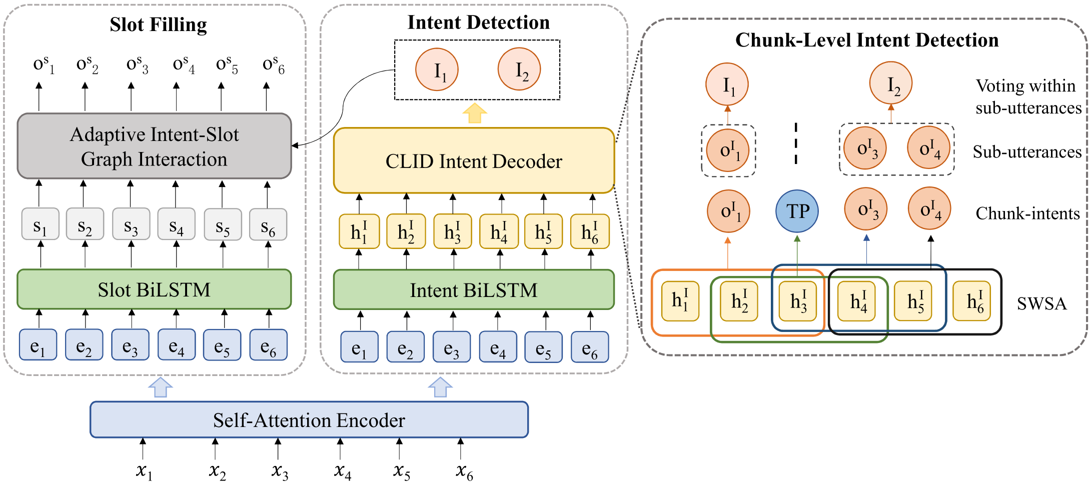

Slot Filling
o^s_1 o^s_2 o^s_3 o^s_4 o^s_5 o^s_6
Adaptive Intent-Slot Graph Interaction
s_1 s_2 s_3 s_4 s_5 s_6
Slot BiLSTM
e_1 e_2 e_3 e_4 e_5 e_6
Intent Detection
I_1 I_2
CLID Intent Decoder
h^I_1 h^I_2 h^I_3 h^I_4 h^I_5 h^I_6
Intent BiLSTM
e_1 e_2 e_3 e_4 e_5 e_6
Self-Attention Encoder
x_1 x_2 x_3 x_4 x_5 x_6
Chunk-Level Intent Detection
I_1 I_2
Voting within sub-utterances
o^I_1 o^I_3 o^I_4
Sub-utterances
o^I_1 TP o^I_3 o^I_4
Chunk-intents
h^I_1 h^I_2 h^I_3 h^I_4 h^I_5 h^I_6
SWSA

Fig. 2. Illustration of our proposed framework for multiple intent SLU.

regional context information, and thus fail to recognize the correct intent.

In this paper, to address the above issues, we propose a Chunk-Level Intent Detection (CLID) framework, which offers proper-grained semantic information integration for intent recognition, as shown in Fig. 1(b). In practice, we introduce a sliding window-based self-attention (SWSA) scheme for regional intent detection, which can capture the contextual information of each token within a chunk and hence predict the intent label for each chunk. For the first issue, we introduce an auxiliary task based on SWSA to predict the intent transition point in an utterance. For the second issue, we obtain sub-utterances with a single intent, based on the identified transition point. The intent of a sub-utterance is computed by voting from the predicted intent of each chunk within the sub-utterance. The chunk-level prediction with sliding windows can avoid the problem of missing necessary information caused by utterance-level intent detection and the problem of missing context information for token-level intent detection. For the slot filling part, we implement the slot decoder part from the AGIF model [12]. Experimental results on two public datasets MixATIS and MixSNIPS show that our framework outperforms the state-of-the-art methods.

II. Proposed Model

Our proposed model shown in Fig. 2 consists of a shared self-attention encoder (§ II.A), a chunk-level intent detection (CLID) decoder (§II.B), and a slot filling decoder (§II.C). Both intent detection and slot filling are jointly trained via multi-task learning (§II.D). The proposed CLID primarily involves: (i) sliding window-based self-attention (SWSA) scheme; (ii) chunk-intent detection and intent transition point (TP) identification; and (iii) sub-utterance intent prediction.

A. Self-Attention Encoder

Given an utterance with a sequence of token $\{t_1, t_2, \ldots, t_n\}$, the input embedding layer ϕ^{emb} maps the token sequences into a sequence of embedding $\boldsymbol{X} = \{\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_n}\} \in \mathbb{R}^{n \times d}$ (d indicates the embedding dimension). Following Qin et al. [12], a self-attention encoder with bidirectional LSTM (BiLSTM) is leveraged to capture the features within token orders and contextual information. The BiLSTM [16] generate the contextual sensitive hidden states $\boldsymbol{H} = \{\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_n\}$ by using the $\boldsymbol{h_i} = BiLSTM(\boldsymbol{x_i})$. Inspired by Vaswani et al. [17], the self-attention mechanism is utilized over a token representation matrix. $\boldsymbol{A} = SelfAttention(\boldsymbol{H})$. $\boldsymbol{H}$ and $\boldsymbol{A}$ are concatenated into a matrix:

$$\boldsymbol{E} = [\boldsymbol{H}; \boldsymbol{A}] \in \mathbb{R}^{n \times 2d}, \quad (1)$$

where $\boldsymbol{E} = \{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_n\}$, and is fed into the decoder as final encoding representation.

B. Chunk-Level Intent Detection

The core contribution of this paper is the chunk-level intent detection to relieve the problem caused by token- and utterance-level intent prediction. We propose a SWSA scheme to capture the contextual information within a chunk for regional intent detection. Based on the SWSA, we predict the transition point at each utterance to cut the utterance into sub-utterances with a single intent. The intent of each sub-utterance is then predicted by assembling the intent predictions of the chunks (in a sliding window manner) within it.

Specifically, $\boldsymbol{E}$ is fed into a intent BiLSTM to promote its task-specific representation:

$$\boldsymbol{h}_t^I = BiLSTM(\boldsymbol{e}_t, \boldsymbol{h}_{t-1}^I, \boldsymbol{h}_{t+1}^I), \quad (2)$$

1) Sliding Window-Based Self-Attention: In the SWSA scheme, a window is used to slide through the utterance, and we do self-attention in the window. Then, $\boldsymbol{h}_t^I$ is fed in window-self-attention to calculate the $\boldsymbol{H}_I^{win} = \{\boldsymbol{h}_1^{win}, \ldots, \boldsymbol{h}_w^{win}\}$, where w denotes the number of the window:

$$\boldsymbol{A}_t = softmax\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{d_k}}\right)\boldsymbol{V}, \quad (3)$$

$$\boldsymbol{h}_t^{win} = \sum_{i=1}^{win\_size} \boldsymbol{a}_i, \quad (4)$$

where $\boldsymbol{H}_t^I = \{\boldsymbol{h}_t^I, \ldots, \boldsymbol{h}_{t+win\_size}^I\}$ is the matrix framed by a window, $\boldsymbol{A}_t = \{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_i, \ldots, \boldsymbol{a}_{win\_size}\}$ is the matrix computed by the self-attention, and the matrix $\boldsymbol{H}_t^I \in \mathbb{R}^{win\_size \times 2d}$ is mapped to queries $\boldsymbol{Q}$, keys $\boldsymbol{K}$ and values $\boldsymbol{V}$ by utilizing the

TABLE I

SLU Performance on Two Datasets. Reproducing Results for GL-GIN are Labeled With †. The Numbers With * Indicate That the Improvement of Our Model Over All Baselines is Statistically Significant With $p < 0.05$ Under T-Test

Model	MixATIS Dataset			MixSNIPS Dataset		
	Intent(Acc)	Slot(F1)	Overall(Acc)	Intent(Acc)	Slot(F1)	Overall(Acc)
Attention BiRNN	74.6	86.4	39.1	95.4	89.4	59.5
Slot-Gated	63.9	87.7	35.5	94.6	87.9	55.4
Bi-Model	70.3	83.9	34.4	95.6	90.7	63.4
SF-ID Network	66.2	87.4	34.9	95.0	90.6	59.9
Stack-Propagation	72.1	87.8	40.1	96.0	94.2	72.9
Joint Multiple ID-SF	73.4	84.6	36.1	95.1	90.6	62.9
AGIF	74.4	86.7	40.8	95.1	94.2	74.2
GL-GIN †	75.6	87.2	41.6	95.2	93.7	72.4
CLID (ours)	**77.5***	**88.2***	**49.0***	**96.6***	**94.3**	**75.0***

distinct linear projection parameters $\boldsymbol{W_q}, \boldsymbol{W_k}, \boldsymbol{W_v}$. The inner product between $\boldsymbol{Q}$ and $\boldsymbol{K}$ is used to calculate the attention weight. The production of self-attention $\boldsymbol{A}_t \in \mathbb{R}^{win\_size \times 2d}$ is a weighted sum of $\boldsymbol{V}$.

2) Chunk-Intent Detection and Intent Transition Point Identification: Based on the SWSA, $\boldsymbol{h}_t^{win}$ is used for the chunk-intent detection at the t-th window:

$$\boldsymbol{y}_t^I = \sigma(\boldsymbol{W}_I(LeakyReLU(\boldsymbol{W}_h \boldsymbol{h}_t^{win} + \boldsymbol{b}_h)) + \boldsymbol{b}_I), \tag{5}$$

$$o_t^I = argmax(\boldsymbol{y}_t^I), \tag{6}$$

where o_t^I is the predicted intent label at the t-th window; σ represents the sigmoid activation function; $\boldsymbol{W}_h$ and $\boldsymbol{W}_I$ are the trainable parameters. $\boldsymbol{b}_h$ and $\boldsymbol{b}_I$ denote bias.

An auxiliary task is introduced to identify the intent transition point (TP) in an utterance. Qin et al. [12] construct the multi-intent datasets by concatenating multiple utterances with conjunctions so that we can locate the conjunction easily and label the conjunctions as "TP" for training. The "TP" is regarded as another intent label. Then, based on the identified transition point, we can separate the utterance into several single-intent sub-utterances.

3) Sub-Utterance Intent Prediction: The intent of i-th sub-utterance I_i is generated by voting from the predicted chunk-intents (in a sliding window manner) within sub-utterance i:

$$I_i = argmax \sum_{t=1}^{l_i} \sum_{k=1}^{n_I} \alpha_k \mathbb{1}[o_t^I = k], \tag{7}$$

where l_i is the number of chunks within i-th sub-utterance and n_I denotes the number of intent labels; α_k is a 0-1 vector $\alpha \in \mathbb{R}^{n_I}$ of which k-th unit is 1, and the others are 0.

The final utterance intent result $\boldsymbol{I} = \{I_1, \ldots, I_s\}$ is the collection of the intent of sub-utterances, where s denotes the number of sub-utterances.

C. Slot Filling Decoder

We implement the same slot filling decoder as AGIF [12] for its promising performance on the interaction between slots and multiple intents. The predicted multiple intents information $\boldsymbol{I} = \{I_1, \ldots, I_s\}$ is utilized to guide the t-th slot prediction:

$$\boldsymbol{y_t^s} = Slot\_Decoder(\boldsymbol{e_t}, \boldsymbol{I}), \tag{8}$$

$$o_t^s = argmax(\boldsymbol{y_t^s}), \tag{9}$$

where o_t^s is the predicted slot for t-th token. For more details on the slot filling decoder, please refer to [12].

D. Multi-Task Training

Considering the correlation between two sub-tasks, we train our model jointly. The chunk-level intent detection objective is formulated as:

$$\mathcal{L}_{intent} = -\sum_{i=1}^{n} \sum_{j=1}^{n_I} \hat{y_i}^{(j,I)} log(y_i^{(j,I)}), \tag{10}$$

where the n_I is the number of the intent and $\hat{y_i}^{(j,I)}$ is the gold intent label.

Similarly, the task objective of slot filling is formulated as:

$$\mathcal{L}_{slot} = -\sum_{i=1}^{n} \sum_{j=1}^{n_S} \hat{y_i}^{(j,S)} log(y_i^{(j,S)}), \tag{11}$$

where the n_S is the number of the slot and $\hat{y_i}^{(j,S)}$ is the gold slot label. The final joint objective is:

$$\mathcal{L} = \alpha \mathcal{L}_{intent} + (1 - \alpha)\mathcal{L}_{slot}, \tag{12}$$

where α is a hyper-parameter.

III. Experiment

A. Experimental Setup

1) Datasets: We evaluate the proposed framework on two publicly available multi-intent datasets. One is MixATIS [12], [13], [18] with $13162, 756, 828$ utterances for training, validation and testing. Another is MixSNIPS [12], [13], [19] with 39776, 2198, 2199 utterances for training, validation and testing. For both MixATIS and MixSNIPS, the percentages of utterances with 1-3 intents are 30%, 50% and 20% [12], [13].

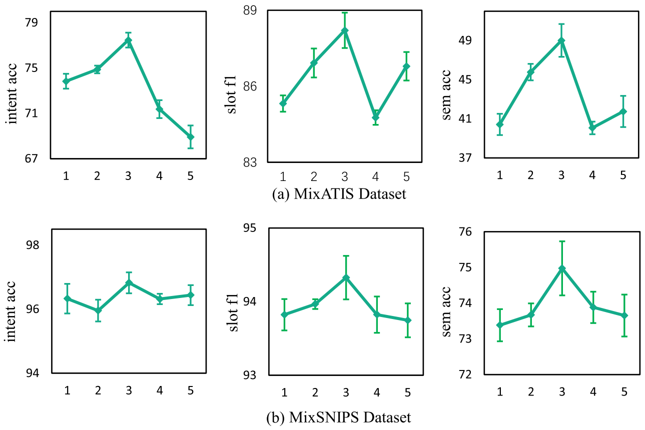
2) Implementation Details: Adam [20] is used for optimizing the parameters of our model. The embedding size is set to 128 and 32 on MixATIS and MixSNIPS, respectively. The dimensionality of the LSTM hidden units is set to 256. The window size is 3. The number of graph attention networks is set to 2. The hyper-parameters are tuned using the validation set. In our experiments, we select the average of 10 times independent experiments as result.

3) Baselines: We compare the proposed *CLID* with the following baselines in three groups. (1) State-of-the-art single-intent SLU models including *Attention BiRNN* [4], *Slot-Gated* [5], *Bi-Model* [7], *SF-ID Network* [8], and *Stack-Propagation* [9]. (2) Multiple intent SLU models with utterance-level intent detection framework including *Joint Multiple ID-SF* [11] and *AGIF* [12]. (3) Multiple intent SLU models with token-level intent detection framework, *GL-GIN* [13].

TABLE II

THE RESULT COMES FROM THE DATASET MIXATIS. THE **INTENT NUM** DENOTES THE NUMBER OF INTENTS IN AN UTTERANCE

Model	**intent num = 1**			**intent num = 2**			**intent num = 3**		
	Intent(Acc)	Slot(F1)	Overall(Acc)	Intent(Acc)	Slot(F1)	Overall(Acc)	Intent(Acc)	Slot(F1)	Overall(Acc)
AGIF	90.4	88.4	73.2	71.6	86.9	36.4	56.3	86.4	21.1
GL-GIN	91.3	88.0	72.6	76.2	87.3	39.1	63.1	86.8	23.0
CLID (ours)	**94.7**	**88.6**	**76.4**	**77.5**	**88.1**	**48.4**	**64.3**	**87.6**	**28.5**



(a) MixATIS Dataset

(b) MixSNIPS Dataset

Fig. 3. Effect of different window sizes of the sliding window in the proposed SWSA scheme on two datasets.

In this section, we demonstrate the effectiveness of the proposed *CLID* from several perspectives. We answer the following research questions (RQs) in the experiments:

- **RQ1:** What is the performance of *CLID* as compared with baseline models?
- **RQ2:** How does the window size influence the performance of the proposed *CLID*?
- **RQ3:** Where do improvements of *CLID* come from?

B. RQ1: Overall Comparisons

We compare the multi-intent SLU performances of all models in Table I. We run the published code of *GL-GIN* and the re-produced results are labeled with †, and the rest result of compared models are taken from [13]. Following Goo et al. [5] and Qin et al. [12], [13], we utilize Intent(Acc) and Slot(F1) to evaluate the intent detection and slot filling sub-task, respectively, and use Overall(Acc) to evaluate the utterance-level performance, which represents both slot filling and intent detection are correct in an utterance. Because the proposed *CLID* focuses on improving intent detection, we mainly observe the performance of Intent(Acc) and Overall(Acc) and also take into account the Slot(F1).

From the result in Table I, we have the following observations: (1) *CLID* achieves the best performance against all baselines in all metrics, which demonstrates the superior multi-intent SLU performance of *CLID* over existing methods. (2) Our model outperforms the token-level intent detection model *GL-GIN*† by 1.9% on Intent(Acc) and 7.4% on Overall(Acc) on the MixATIS dataset. Besides, we achieves 1.4% and 2.6% improvements on Intent(Acc) and Overall(Acc) on MixSNIPS dataset. (3) Compared to the utterance-level intent detection model *AGIF* that uses the same slot filling decoder as us, the Slot(F1) performance gained of *CLID* is 1.5% and 0.1% on MixATIS and MixSNIPS datasets, which indicates that the performance of slot filling also benefits from the improvement of intent detection.

C. RQ2: Effectiveness of the Window Size

In this section, we analyze the influence of the critical parameter, that is, the window size of the sliding window in the proposed SWSA scheme. The performance of the proposed *CLID* with a varying window size on MixATIS and MixSNIPS datasets is illustrated in Fig. 3. We can observe that the performance of all metrics first increases and then drops as the value of window size grows. The particular case is that the window size = 1 as an ablation study, where the SWSA scheme does not affect the proposed *CLID*. We can see that the performance drops significantly when window size = 1, from which we can conclude that the chunk-level semantics is crucial for improving multi-intent detection. Moreover, further increasing the window size might still hurt the performance. The potential reason may be that the performance of the intent transition identification may drop with a too-long chunk.

D. RQ3: Improvement Analysis

In this section, we analyze the origins of improvements of *CLID* by investigating the performance differences in groups of intent numbers. We classify the utterances in the test set by the number of intents they carried, then put them into our model respectively to evaluate our model. The results are shown in Table II. The results of *AGIF* and *GL-GIN* in different intent groups are reproducing results. From Table II, we can observe that: (1) The performance improvement of our model on multiple intents utterances is significant. We gain 1.92%, 0.94% and 23.91% relative improvement on Intent(Acc), Slot(F1) and Overall(Acc) compared with *GL-GIN* in 3-intents utterances. (2) The relative improvement in single intent utterances is also significant. Compared with *GL-GIN*, *CLID* achieve 3.66%, 0.62% and 5.19% relative improvements on Intent(Acc), Slot(F1) and Overall(Acc).

IV. CONCLUSION

In this paper, we propose a chunk-level intent detection (CLID) framework to alleviate the problem that is too coarse-grained at utterance-level as well as too fine-grained at token-level. In CLID, we propose a sliding window-based self-attention (SWSA) scheme and introduce an auxiliary task of intent transition point identification to adapt to the characteristic of multiple intents utterances. By identifying the intent transition point in an utterance and obtaining sub-utterances with a single intent, our model can correctly recognize the intents of the sub-utterances. Experiments on two public multi-intent datasets show the effectiveness of the proposed model.

REFERENCES

[1] G. Tur and R. D. Mori, *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech*. New York, NY, USA: Wiley, 2011. [Online]. Available: https://doi:10.2200/S00134ED1V01Y200807SAP00

[2] K. Yao, B. Peng, G. Zweig, D. Yu, X. Li, and F. Gao, "Recurrent conditional random field for language understanding," in *Proc. IEEE 39th Int. Conf. Acoust., Speech, Signal Process.*, 2014, pp. 4077–4081.

[3] K. Yao, B. Peng, Y. Zhang, D. Yu, G. Zweig, and Y. Shi, "Spoken language understanding using long short-term memory neural networks," in *Proc. IEEE Workshop Spoken Lang. Technol.*, 2014, pp. 189–194.

[4] B. Liu and I. Lane, "Attention-based recurrent neural network models for joint intent detection and slot filling," in *Proc. 17th Annu. Conf. Int. Speech Commun. Assoc.*, 2016, pp. 685–689.

[5] C. Goo et al., "Slot-gated modeling for joint slot filling and intent prediction," in *Proc. 16th Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2018, pp. 753–757.

[6] C. Li, L. Li, and J. Qi, "A self-attentive model with gate mechanism for spoken language understanding," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 3824–3833.

[7] Y. Wang, Y. Shen, and H. Jin, "A bi-model based RNN semantic frame parsing model for intent detection and slot filling," in *Proc. Conf. 16th North Amer. Chapter Assoc. Computat. Linguistics: Hum. Lang. Technol.*, New Orleans, Louisiana, USA, 2018, pp. 309–314. [Online]. Available: https://doi.org/10.18653/v1/n18-2050

[8] H. E. et al., "A novel bi-directional interrelated model for joint intent detection and slot filling," in *Proc. 57th Conf. Assoc. Comput. Linguistics*, 2019, pp. 5467–5471.

[9] L. Qin et al., "A stack-propagation framework with token-level intent detection for spoken language understanding," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process.*, 2019, pp. 2078–2087.

[10] Z. Zhu, P. Huang, H. Huang, S. Liu, and L. Lao, "A graph attention interactive refine framework with contextual regularization for jointing intent detection and slot filling," in *Proc. IEEE 47th Int. Conf. Acoust., Speech, Signal Process.*, 2022, pp. 7617–7621. [Online]. Available: https://doi.org/10.1109/ICASSP43922.2022.9746942

[11] R. Gangadharaiah and B. Narayanaswamy, "Joint multiple intent detection and slot labeling for goal-oriented dialog," in *Proc. 17th Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, Minneapolis, MN, USA, 2019, pp. 564–569. [Online]. Available: https://doi.org/10.18653/v1/n19-1055

[12] L. Qin, X. Xu, W. Che, and T. Liu, "Towards fine-grained transfer: An adaptive graph-interactive framework for joint multiple intent detection and slot filling," in *Proc. Findings Assoc. Comput. Linguistics: EMNLP, Online Event, 16-20 November, Ser. Findings ACL*, 2020, pp. 1807–1816. [Online]. Available: https://doi.org/10.18653/v1/2020.findings-emnlp.163

[13] L. Qin, F. Wei, T. Xie, X. Xu, W. Che, and T. Liu, "GL-GIN: Fast and accurate non-autoregressive model for joint multiple intent detection and slot filling," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics, 11th Int. Joint Conf. Natural Lang. Process.*, 2021, pp. 178–188. [Online]. Available: https://doi.org/10.18653/v1/2021.acl-long.15

[14] P. Xu and R. Sarikaya, "Exploiting shared information for multi-intent natural language sentence classification," in *Proc. 14th Annu. Conf. Int. Speech Commun. Assoc.*, 2013, pp. 3785–3789. [Online]. Available: http://www.isca-speech.org/archive/interspeech_2013/i13_3785.html

[15] B. Kim, S. Ryu, and G. G. Lee, "Two-stage multi-intent detection for spoken language understanding," *Multimedia. Tools Appl.*, vol. 76, no. 9, pp. 11377–11390, 2017. [Online]. Available: https://doi.org/10.1007/s11042-016-3724-4

[16] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[17] A. Vaswani et al., "Attention is all you need," in *Proc. 31st Annu. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008. [Online]. Available: https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

[18] C. T. Hemphill, J. J. Godfrey, and G. R. Doddington, "The ATIS spoken language systems pilot corpus," in *Proc. Speech Natural Lang.: Workshop Held Hidden Valley*, 1990, pp. 96–101. [Online]. Available: https://aclanthology.org/H90-1021/

[19] A. Coucke et al., "Snips voice platform: An embedded spoken language understanding system for private-by-design voice interfaces," 2018. [Online]. Available: http://arxiv.org/abs/1805.10190

[20] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Representations*, San Diego, CA, USA, 2015.