Multi-intent Attention and Top-k Network with Interactive Framework for Joint Multiple Intent Detection and Slot Filling

Xu Jia(✉), Jiaxin Pan, Youliang Yuan, and Min Peng(✉)

School of Computer Science, Wuhan University, Wuhan, China
{jia_xu,pjx_1997,2020282110194,pengm}@whu.edu.cn

Abstract. Multiple intent detection and slot filling are essential components of spoken language understanding. Existing methods treat multiple intent detection as a multi-label classification task. However, multi-label classification methods focus on the correlation between different intents and set the threshold to select the high probability intents. These methods will cause the model to miss part of the correct intents. In this paper, to address this issue, we introduce Multi-Intent Attention and Top-k Network with Interactive Framework (MIATIF) for joint multiple intent detection and slot filling. In particular, we model the multi-intent attention to obtaining the relation between the utterance and intents. Meanwhile, we propose the top-k network to encode the distribution of different intents and accurately predict the number of intents. Experimental results on two publicly available multiple intent datasets show substantial improvement. In addition, our model saves 64%–72% of training time compared to the current state-of-the-art graph-based model.

Keywords: Interactive framework · Multiple intent detection · Multi-intent attention · Top-k network

1 Introduction

Intent detection and slot filling are significant parts of spoken language understanding [13]. In an utterance, intents and slots always exist a strong correlation. For instance, the slot of movie name "*paris by night*" and the intent "*SearchScreeningEvent*" correspond to each other in the query "*Rate if tomorrow comes and what time will paris by night aired*". To model the relation between intents and slots, dominant models [4,5,12,16,23] adopt joint models to build the relationship between the two tasks. Though achieving promising performances, previous works only focus on the single-intent task. However, the utterances in reality dialogue scenarios express more than a single intent [3]. For example, in Fig. 1, the whole sentence corresponds to the intent "*RateBook*" and the intent "*SearchScreeningEvent*".

L. Wang et al. (Eds.): NLPCC 2021, LNAI 13028, pp. 467–479, 2021.
https://doi.org/10.1007/978-3-030-88480-2_37

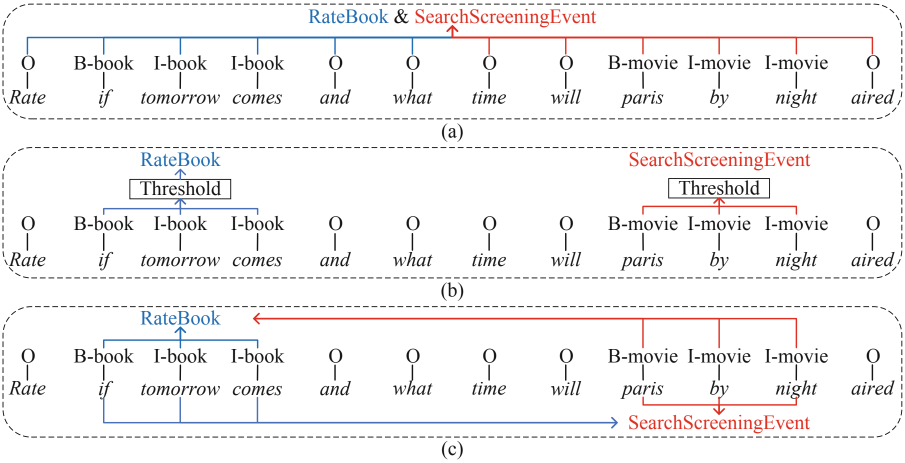
RateBook & SearchScreeningEvent
O B-book I-book I-book O O O O B-movie I-movie I-movie O
Rate if tomorrow comes and what time will paris by night aired
(a)
RateBook
Threshold
SearchScreeningEvent
Threshold
(b)
RateBook
SearchScreeningEvent
(c)

Fig. 1. Prior works treat multiple intents as an entire intent (a) or use multi-label classification methods to filter intents under the threshold (b). Our method discards the threshold and uses the attention between the utterance and intents to determine the final multiple intents (c).

To solve the problem of multiple intents in an utterance, the prior models directly combine multiple intents into a single one, which is shown in Fig. 1(a). However, these models do not guide each word to capture the features corresponding to different intents [17]. To better perform multiple intent detection and slot filling, [3] and [17] achieve promising performance by using multi-label classification methods to consider two tasks jointly, as shown in Fig. 1(b). The multi-label classification methods mainly utilize latent relevance among labels. However, the core of multiple intent detection is to distinguish the irrelevance of different intents. In addition, the method of setting the threshold can only select intents with higher probability. Depicted in Fig. 1(b), the intent "*RateBook*" which is above the threshold can be selected. In Fig. 1(c), these tokens "*paris by night*" not only focus on the intent "*SearchScreeningEvent*", but also reduce the relevance on the intent "*RateBook*".

There are two challenges in multiple intent detection: 1) How to distinguish the features of different intents. 2) How to predict the number of multiple intents rather than setting the threshold. To solve these two problems, we propose a **M**ulti-**I**ntent **A**ttention and **T**op-k Network with **I**nteractive **F**ramework (MIATIF). In particular, we use an interactive framework based on the vanilla transformer to improve the performance of both multiple intent detection and slot filling. We introduce multi-intent attention to capture the relation between the utterance and intents, which helps distinguish different intents' features. Meanwhile, we construct the top-k network to predict the number of intents by encoding the distribution of different intents. This network can replace the method of setting a threshold to avoid missing low probability intents.

To summarize, the contributions of this paper are: 1) We propose a Multi-Intent Attention and Top-k Network with Interactive Framework to jointly solve

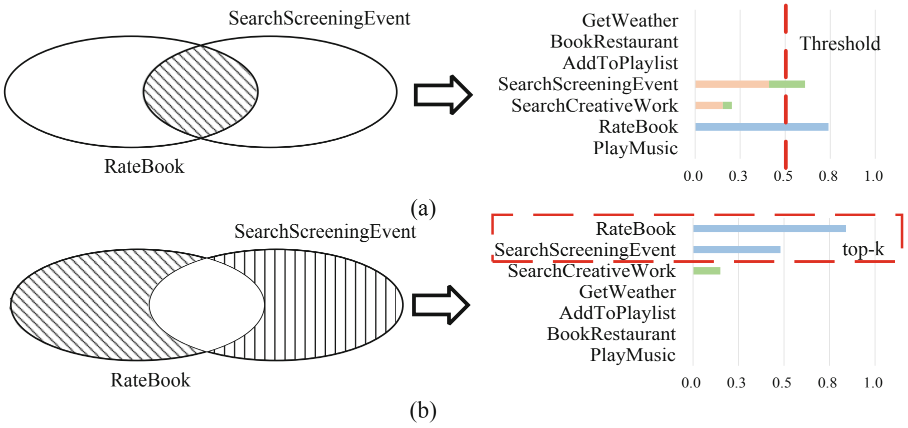

Fig. 2. (a) Multi-label classification methods set the threshold to select high probability labels. (b) Multiple intent detection methods pay more attention to the difference of intents.

the problem of multiple intent detection and slot filling. 2) We introduce multi-intent attention to distinguish the features between different intents. The top-k network predicts the number of intents by encoding the distribution of different intents. 3) We evaluate the performance of our model on two publicly available dialogue datasets. Our model shows improve overall accuracy performance 3.1% and 1.3% on two datasets and save 64%–72% training time compared to the current state-of-the-art method.

2 Problem Definition

Current works treat multiple intent detection as a multi-label classification task. Models select the intents which have high probability by setting the threshold. However, this paper argues that multiple intent detection and multi-label classification are essentially different tasks. As shown in Fig. 2(a), multi-label classification exploits the association between two labels to improve label probability to avoid being filtered by the threshold. In most cases, there is not a strong correlation between the different intents in an utterance. Therefore, the model needs to focus on the features between different intents in multiple intent detection. In this paper, we redefine the task of multiple intent detection.

We define an utterance $U = (w_1, w_2, \ldots, w_L)$ consists of a sequence of L words. Multiple intent detection needs to decide the multiple intent label $Y^I = (Y^I_1, \ldots, Y^I_{\hat{k}})$ with $\hat{k}$ possible intents. We should learn a function $f_I : U \rightarrow Y^I$ from sufficient training samples that achieve the mapping from utterance to multiple intents. In most multi-label classification models, $f_I(U) = \{Y^I | Sim(U, \hat{Y}^I) > \delta\}$ will be derived, where $Sim(U, \hat{Y}^I)$ evaluate the relevance scores of all intents and utterance, and δ is the threshold value. In multiple intent detection, we first learn a top-k function $f_k : U \rightarrow k$, which utilizes the representation of the different intents in the utterance to predict the number of intents. We set the function $f_I(U) = \{Y^I | Top(Sim(U, \hat{Y}^I), f_k(U))\}$, where $Top(Y, k)$ denotes taking the top k values from Y. We train the model to find the best parameter set α that maximizes the likelihood:

$$\arg\max_{\alpha} P(Y^I | f_I(U); \alpha). \tag{1}$$

3 Model

The architecture of our model is shown in Fig. 3, which consists of the multi-intent attention and the top-k network based on the interactive framework.

3.1 Interactive Framework

In single intent detection and slot filling tasks, the interactive framework improves the performance by model the bidirectional connection between the intents and slots [18]. We firstly perform an interactive framework based on the vanilla transformer [20] to multiple intent detection. In the context feature encoder, We adopt the BiLSTM to encode each utterance U to produce a series of hidden states $H = (h_1, h_2, \ldots, h_L)$. We use H^C to represent the output of the context feature encoder. Then, we get the explicit multiple intents and slots representation and put them into the interactive framework to make a mutual interaction. We randomly initialize the parameters as intent embedding matrix $W_F^I \in R^{d \times N_I}$ and slot embedding matrix $W_F^S \in R^{d \times N_S}$ (d represents the dimension of hidden states; N_I and N_S represent the number of intents and slots, respectively).

In practice, we use W_F^I and W_F^S to obtain H^I and H^S, respectively:

$$H^I = H^C + softmax(H^C \cdot W_F^I) \cdot W_F^I, \tag{2}$$

$$H^S = H^C + softmax(H^C \cdot W_F^S) \cdot W_F^S. \tag{3}$$

Furthermore, we map the matrix H^I and H^S to queries (Q^I, Q^S), keys (K^S, K^I) and values (V^S, V^I) by using different linear projections. Finally, we treat Q^S as queries, K^I as keys, and V^I as values and obtain new slot representations incorporating intent information. The new slot representations:

$$\hat{H}^S = H^S + softmax(\frac{Q^S K^I}{\sqrt{d}})V^I. \tag{4}$$

Similarly, we obtain the new intent representations:

$$\hat{H}^I = H^I + softmax(\frac{Q^I K^S}{\sqrt{d}})V^S. \tag{5}$$

The interactive framework enables sharing the features of intents and slots. It can avoid the phenomenon of an utterance with correct slots and wrong intent or correct intent and wrong slots.

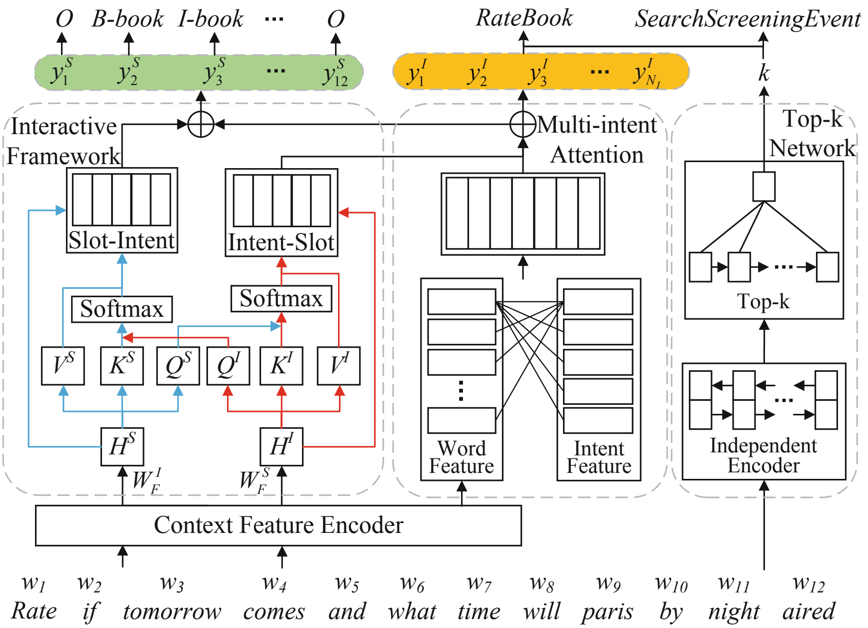

Fig. 3. The illustration of multi-intent attention and top-k network with interactive framework (MIATIF). Multi-intent attention and top-k network distinguish the features of different intents and accurately predict the number of intents.

3.2 Multi-intent Attention and Top-k NetWork

In this paper, the core contribution is the multi-intent attention and the top-k network. Firstly, multi-intent attention can build the relationship between the utterance and intents, distinguishing features of different intents by the text semantics of utterance. Then, to predict the number of intents, we introduce an independent encoder to encode the different distribution features. We take the representations of different intents and the top-k to predict the number of intents in the utterance.

Multi-intent Attention. The text of intents usually has specific semantics [22]. To make use of the semantic information of multiple intents, we need to obtain the intent embedding matrix $E^I \in R^{N_I \times d}$ in the same latent d-dim space with the words.

After obtaining the hidden states H^C from the context feature encoder and the intent embedding E^I, we can explicitly determine the semantic relation between each pair of words and intents. Attention weights are computed by the dot product between H^C, E^I, and output $A^I \in R^{L \times N_I}$:

$$A^I = softmax(H^C \cdot E^I); \hat{A}^I = A^I \cdot E^I, \tag{6}$$

where $\hat{A}^I \in R^{L \times d}$ is the relationship between each pair of words and intents. The representation $\hat{A}^I$ is based on multiple intents and words in an utterance. Thus, we call it multi-intent attention.

Top-k Network. In this paper, one challenge we focus on is how to replace the threshold method to predict the number of multiple intents $\hat{k}$ accurately. We propose the top-k network to accomplish multiple intent detection by encoding the distribution of different intents in an utterance. Unlike the method from [10], we do not bother to predict some conjunctions to determine whether it is a multiple intent problem. Therefore our method has universal application scenarios for predicting the number of intents rather than focusing on some special tokens. In particular, to avoid integrating the context features, we use an independent encoder to encode multi-intent distribution. The same with context feature encoder, we can get H^M from the output of the independent encoder.

Then, we use a unidirectional LSTM as the top-k decoder, which predicts the number of multiple intents. The intent distribution vector H^M will be fed to the decoder to predict the number of multiple intents. At each step i, the decoder state s_i^k is calculated by previous decoder state s_{i-1}^k, the previous number of multiple intents k_{i-1} and the aligned encoder hidden state h_i^M:

$$s_i^k = LSTM(s_{i-1}^k, k_{i-1}, h_i^M); k = \lfloor \sum_{i=1}^{L} (W_i^n \cdot s_i^k + b_i) + \frac{1}{2} \rfloor, \tag{7}$$

where $\lfloor \cdot \rfloor$ indicates round down.

3.3 Decoder

In multi-intent attention and top-k network with the interactive framework, we have obtained intent representation $\hat{H}^I$, slot representation $\hat{H}^S$, the multi-intent attention $\hat{A}^I$, and the number of multiple intents k. In this section, we build an intent decoder and a slot decoder, respectively.

Intent Decoder. We concatenate the intent representation $\hat{H}^I$ and the multi-intent attention $\hat{A}^I$ as the representation of the final inputs:

$$\tilde{H}^I = \hat{H}^I \oplus \hat{A}^I, \tag{8}$$

where $\tilde{H}_I \in R^{T \times 2d}$ and $\oplus$ is an operation for concatenating two vectors.

We use a unidirectional LSTM as the multiple intent detection decoder:

$$s_i^I = LSTM(s_{i-1}^I, y_{i-1}^I, \tilde{h}_i^I). \tag{9}$$

Then the decoder state s_i^I is utilized for multiple intent detection:

$$y^I = \sigma(LeakyReLU(W_1^I s^I + b_1^I) W_2^I + b_2^I), \tag{10}$$

where W_1^I, W_2^I are trainable parameters of the intent decoder, $y^I = \{y_1^I, \ldots, y_{n_I}^I\}$ is the intent output of the utterance and σ represents the activation function.

We use the number of multiple intents k in each utterance during inference instead of setting a threshold. The final result O^I is generated by intent output y^I and the number of multiple intents k. We get the top-k largest intent distributions as the final output. For example, if the $y^I = \{0.7, 0.1, 0.3, 0.9, 0.5, 0.1, 0.2\}$ and the k is 2, we predict intents $O^I = \{1, 4\}$.

Slot Decoder. For the slot filling decoder, we similarly use another unidirectional LSTM as the slot filling decoder. To ensure the performance of the slot filling task, we leverage multiple intent features to guide the slot prediction. At the decoding step i, the decoder state s_i^S can be formalized as:

$$s_i^S = LSTM(s_{i-1}^S, y_{i-1}^S, \hat{h}_i^S \oplus \tilde{h}_i^I). \tag{11}$$

Similarly, the decoder state s_i^S is utilized for slot filling:

$$y_i^S = softmax(W_d^S s_i^S); O_i^S = argmax(y_i^S), \tag{12}$$

where O_i^S is the slot label of the i-th word in the utterance.

3.4 Joint Training

Following [4,16,17], we adapt a joint model to consider the three tasks and update parameters by joint optimizing. The intent detection, slot filling, and top-k loss functions are:

$$\mathcal{L}^I = -\sum_{m=1}^{n_I} (\hat{y}_m^I log(y_m^I) + (1 - \hat{y}_m^I) log(1 - y_m^I)), \tag{13}$$

$$\mathcal{L}^S = -\sum_{i=1}^{n_W} \sum_{j=1}^{n_W} \hat{y}_i^{(j,S)} log y_i^{(j,S)}, \tag{14}$$

$$\mathcal{L}^k = |k - \hat{k}|, \tag{15}$$

where $\hat{y}^I$, $\hat{y}^S$ and $\hat{k}$ are the gold intent label, gold slot label, and the gold number of intents, respectively.

The final joint objective is formulated as:

$$\mathcal{L} = \mathcal{L}^I + \mathcal{L}^S + \mathcal{L}^k. \tag{16}$$

4 Experiments

4.1 Datasets

Since other single intent datasets cannot evaluate multi-intent models, we evaluate the performance of our model on the only two publicly available multiple intent datasets, MixATIS and MixSNIPS. Both datasets are used in our paper following the same format and partition as in [17].

MixATIS and MixSNIPS datasets are collected from the ATIS [6] and SNIPS [2] which are widely used in SLU task, respectively. [17] utilizes conjunctions to connect sentences with different intents. The number of intents in the datasets is no more than 3, and the ratio between 1–3 intents is 3 : 5 : 2. MixATIS has 18000 utterances for training, 1000 utterances for validation, and 10000 utterances for testing. MixSNIPS has 45000 utterances for training, 2500 utterances for validation, and 2500 utterances for testing. In the training set, MixATIS has 17 different intents, and MixSNIPS has 7.

Table 1. Slot filling and intent detection results on two multi-intent datasets

Model	MixATIS				MixSNIPS			
	Slot(F1)	Intent(F1)	Intent(Acc)	Overall	Slot(F1)	Intent(F1)	Intent(Acc)	Overall
Attention BiRNN [12]	86.6	–	71.6	38.7	89.4	–	94.1	62.2
Slot-Gated [4]	88.1	–	65.7	38.9	87.8	–	96	56.5
SF-ID [5]	87.7	–	63.7	36.2	89.6	–	96.3	59.3
Stack-propagation [16]	87.4	79	71.9	41	93.2	97.6	94.6	71.9
Joint multiple ID-SF [3]	87.5	80.6	73.1	38.1	91	98.2	95.7	66.6
AGIF [17]	**88.1**	**81.2**	75.8	44.5	94.5	**98.6**	96.5	76.4
MIATIF	88.0	78.6	**76.0**	**47.6**	**94.6**	**98.6**	**97.1**	**77.7**

4.2 Implementation Details

The encoder and decoder hidden units are 256 and 128 in all datasets, respectively. We use Adam to optimize the parameters in our model and adapt the suggested hyper-parameters for optimization. For all experiments, we pick the model which the sentence-level accuracy works best on the dev set and then evaluate it on the test set. The epochs are 200 and 100, and the dropout rates are 0.3 and 0.4 for MixATIS and MixSNIPS, respectively. Part of the code uses the MindSpore Lite tool [1].

4.3 Main Results

Following [4] and [17], we use Slot(F1), Intent(F1), Intent(Acc) and Overall to evaluate the performance of slot filling, intent detection and sentence-level accuracy. We adopt the top-k network to predict the number of multiple intents, and the results are 98.6% and 99.6% in the MixATIS and MixSNIPS datasets. Table 1 shows the other experimental results of the proposed models on the MixATIS and MixSNIPS datasets. Among the baselines, [4,5,12,16] are the classical model for single intent, [3,17] achieve state-of-the-art on multiple intent.

We have the following observations from the results: 1) Our model outperforms baseline and achieves promising performances. On the MixATIS dataset, our model achieves 0.2% and 3.1% absolute gains on Intent(Acc) and Overall, respectively. On the MixSNIPS dataset, our model achieves the best results on all metrics, where it improves 0.6% on Intent(Acc) and 1.3% on Overall. The improvement indicates that our model successfully solves the challenge of multiple intent detection and improves the performance of both tasks. 2) The high accuracy of the number of intents reaching 98% has been shown that the top-k network can be relatively reliable. So it ensures that our model will not filter the part of correct intents and only select high probability intents. 3) Compared to the improvement in Intent(Acc), the improvement in Overall is more significant on both datasets. It is because we select the model which has the best performance of Overall on the dev sets. Also, we use the interactive framework to make the sentence-level accuracy perform better by fully interacting with the features of slots and intents in the utterances. 4) The improvements of our model on the Slot(F1) and Intent(F1) are not significant. The reason is that *AGIF* extracts intent features and builds a graph structure to guide slot filling. Meanwhile, they use the threshold to select high probability intents, resulting in higher Intent(F1). However, the graph structure is time-consuming. We use the multi-intent attention to obtain an acceptable slight decrease of Slot(F1) while improving Intent(Acc) and Overall performance.

Table 2. Comparison of training time

Model	MixATIS		MixSNIPS	
	Epoch(s)	All(h:m:s)	Epoch(s)	All(h:m:s)
AGIF	207.5	5:45:50	473.3	6:34:27
MIATIF	74.2	4:07:10	131.5	3:39:14

To show the efficiency of our model, we compare training time with *AGIF*. Table 2 shows the results on the two datasets, where Epoch(s) indicates the average seconds consumed in one epoch and All(h:m:s) represents the time required to complete the full training. As every epoch, our model saves 64% and 72% time consumption, respectively. Although our epoch is twice of *AGIF*, our model still saves 29%–44% of the total training time consumption.

4.4 Ablation Study

In this section, we set up the following ablation experiments to study the impact of our model. The result is shown in Table 3.

Effectiveness of Interactive Framework. For the first time, we apply the interactive framework from single intent detection to multiple intent detection. To verify the validity of the framework, we remove the interactive framework from the model and replace H^I and H^S with H^C. It means that we only get the context feature from the encoder and directly input it into the decoder without incorporating the features of intents and slots. We name it as *without interaction*. From the result, Slot(F1) performances both drop 1.0%, and Intent(Acc) performance drops 0.6% and 0.8%. It results in overall performances drop of 4.4% and 2.5%. The decline Overall is significant without the interactive framework, indicating that the interactive framework plays a key role in sentence-level accuracy. Slot(F1) drops significantly due to the lack of intent features. We introduce multi-intent attention, so Intent(Acc) decreases insignificantly. It verifies that incorporating the intent and slot features is useful for improving the performance of both two tasks.

Effectiveness of Multi-intent Attention. We remove the multi-intent attention and utilize the output H^I of the interactive framework to the intent decoder. We name it as *without multi-intent attention*. From the result, Overall performances both drop 2.9% on the two datasets. We believe the main reason is

Table 3. Ablation experiments on the MixATIS and MixSNIPS datasets

Model	MixATIS				MixSNIPS			
	Slot(F1)	Intent(F1)	Intent(Acc)	Overall	Slot(F1)	Intent(F1)	Intent(Acc)	Overall
W/o interactive	87.0	78.1	75.4	43.2	93.6	98.1	96.3	75.2
W/o multi-intent attention	87.1	78.5	74.6	44.7	94.0	98.1	96.2	74.8
W/o top-k network	87.8	**80.3**	74.7	44.2	94.5	**98.7**	96.7	75.9
MIATIF	**88.0**	78.6	**76.0**	**47.6**	**94.6**	98.6	**97.1**	**77.7**

the decline in Intent(Acc). Since the lack of multi-intent attention, the model cannot distinguish features between different intents. Also, the interactive framework will pass the error to the slot filling, which leads to the decline of Slot(F1) slightly.

Effectiveness of Top-k Network. Instead of adopting the top-k network, we utilize the threshold to predict the multiple intents. We define it as *without top-k network*. This structure is similar with [3] and [17], which perform the multiple intent detection as the multi-label classification. From the result, we observe the overall performances drop 2.4% and 1.8% on the two datasets. We attribute it to the fact that the top-k network can avoid missing useful features of intents. Meanwhile, we observe that the Intent(F1) improves on both datasets due to threshold replacement. Since setting threshold only selects high probability intents, it leads to higher performance on Intent(F1).

5 Related Work

In the current works, intent detection(ID) is usually considered a classification task and slot filling(SF) as a sequence labeling task. So traditional machine learning methods are often used on these two tasks [9,19]. In recent years, various neural architectures have achieved the state-of-the-art [5,7,12,13,15,21]. Due to the strong correlation between the two tasks, the joint model is the currently effective method. The initial works use loss function via backpropagation to verify the parameter of the sharing encode module [12,24]. The later models utilize the features of intent detection to enhance the features of slot filling [4,11,16], and establish the connection between the two tasks using gate mechanism or graph structure [5,14,18].

Although the above joint models have handled both tasks simultaneously, the current single-intent scenario cannot represent the multi-intent scenario. [3] proposes the task of multiple intent detection and introduces the slot-gated mechanism based on token-level to capture the features between intents and slots. To push forward the research of multi-intent SLU, [17] releases two large-scale multi-intent datasets MixATIS and MixSNIPS, based on ATIS and SNIPS. Then [17] introduces an intent-slot graph construction to model the relation between multi-intent and slot filling tasks. Previous works treat multiple intent detection as a multi-label classification task and achieve the promising performance [3,8,17]. Therefore, the above works ignore the differences between different intents. And the threshold only selects intents with high probability. This paper introduces the multi-intent attention and the top-k network to accomplish multiple intent detection and slot filling tasks jointly.

6 Conclusion and Future Work

In this paper, we propose Multi-Intent Attention and Top-k Network with Interactive Framework (MIATIF) for joint multiple intent detection and slot filling. Our model first introduces an interactive framework based on the vanilla transformer in multiple intent detection. Then, to better exploit the features of different intents, we propose multi-intent attention. Furthermore, we utilize the independent encoder to alleviate the mixed context features on multiple intents, and the top-k predicts the number of intents. Our model improves performance for overall accuracy on the MixATIS and MixSNIPS of 3.1% and 1.3%, respectively. Simultaneously, our model saves 64%–72% of training time compared to the current state-of-the-art model while achieving better results. In the future, we also want to introduce pre-trained models to improve the performance using MindSpore.

Acknowledgment. We thank anonymous reviewers for their precious comments. This research is supported by MindSpore, the National Key R&D Program of China under Grant No. 2018YFC1604003, General Program of Natural Science Foundation of China (NSFC) under Grant No. 61772382 and No. 62072346, Key R&D Project of Hubei Province under Grant No. 2020BAA021 and Science and Technology Plan of Wuhan under Grant No. 2020010601012168.

References

1. Mindspore. https://www.mindspore.cn/ (2020)
2. Coucke, A., et al.: Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. arXiv preprint arXiv:1805.10190 (2018)
3. Gangadharaiah, R., Narayanaswamy, B.: 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), pp. 564–569 (2019)
4. Goo, C.W., et al.: 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), pp. 753–757 (2018)
5. Haihong, E., Niu, P., Chen, Z., Song, M.: A novel bi-directional interrelated model for joint intent detection and slot filling. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL), pp. 5467–5471 (2019)
6. Hemphill, C.T., Godfrey, J.J., Doddington, G.R.: The ATIS spoken language systems pilot corpus. In: Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania (1990)

7. Hou, Y., et al.: Few-shot slot tagging with collapsed dependency transfer and label-enhanced task-adaptive projection network. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL), pp. 1381–1393 (2020)
8. Hou, Y., et al.: Few-shot learning for multi-label intent detection. In: The Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI) (2021)
9. Huang, J., et al.: A probabilistic method for emerging topic tracking in microblog stream. World Wide Web (WWW) **20**(2), 325–350 (2017)
10. Kim, B., Ryu, S., Lee, G.G.: Two-stage multi-intent detection for spoken language understanding. Multimed. Tools Appl. **76**(9), 11377–11390 (2017)
11. Li, C., Li, L., Qi, J.: A self-attentive model with gate mechanism for spoken language understanding. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 3824–3833 (2018)
12. Liu, B., Lane, I.: Attention-based recurrent neural network models for joint intent detection and slot filling. In: Proceedings of the 17th Annual Conference of the International Speech Communication Association (INTERSPEECH), pp. 685–689 (2016)
13. Louvan, S., Magnini, B.: Recent neural methods on slot filling and intent classification for task-oriented dialogue systems: a survey. In: Proceedings of the 28th International Conference on Computational Linguistics (COLING), pp. 480–496 (2020)
14. Peng, H., Shen, M., Jiang, L., Dai, Q., Tan, J.: An interactive two-pass decoding network for joint intent detection and slot filling. In: Zhu, X., Zhang, M., Hong, Yu., He, R. (eds.) NLPCC 2020. LNCS (LNAI), vol. 12431, pp. 69–81. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-60457-8_6
15. Peng, M., et al.: Personalized app recommendation based on app permissions. World Wide Web **21**(1), 89–104 (2018)
16. Qin, L., Che, W., Li, Y., Wen, H., Liu, T.: A stack-propagation framework with token-level intent detection for spoken language understanding. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 2078–2087 (2019)
17. Qin, L., Xu, X., Che, W., Liu, T.: Towards fine-grained transfer: an adaptive graph-interactive framework for joint multiple intent detection and slot filling. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings (EMNLP), pp. 1807–1816 (2020)
18. Qin, L., et al.: A co-interactive transformer for joint slot filling and intent detection. In: ICASSP 2021–2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 8193–8197. IEEE (2021)
19. Raymond, C., Riccardi, G.: Generative and discriminative algorithms for spoken language understanding. In: Eighth Annual Conference of the International Speech Communication Association (INTERSPEECH) (2007)
20. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems (NIPS), pp. 5998–6008 (2017)
21. Wu, J., et al.: Joint learning of word and label embeddings for sequence labelling in spoken language understanding. In: 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), pp. 800–806. IEEE (2019)
22. Xiao, L., Huang, X., Chen, B., Jing, L.: Label-specific document representation for multi-label text classification. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 466–475 (2019)

23. Zhang, C., et al.: Joint slot filling and intent detection via capsule neural networks. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL), pp. 5259–5267 (2019)
24. Zhang, X., Wang, H.: A joint model of intent determination and slot filling for spoken language understanding. In: Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI), vol. 16, pp. 2993–2999 (2016)