A Transformer-based Threshold-Free Framework for Multi-Intent NLU

Anonymous ACL submission

Abstract

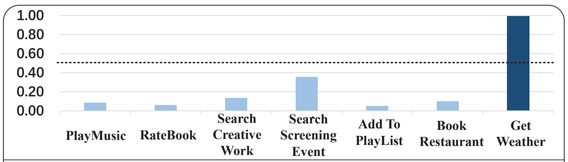
Multi-intent natural language understanding (NLU) has recently gained attention. It detects multiple intents in an utterance, which is better suited to real-world scenarios. However, the state-of-the-art joint NLU models mainly detect multiple intents on threshold-based strategy, resulting in one main issue: the model is extremely sensitive to the threshold settings. In this paper, we propose a transformer-based Threshold-Free Multi-intent NLU model (TFMN) with multi-task learning (MTL). Specifically, we first leverage multiple layers of a transformer-based encoder to generate multi-grain representations. Then we exploit the information of the number of multiple intents in each utterance without additional manual annotations and propose an auxiliary detection task: Intent Number detection (IND). Furthermore, we propose a threshold-free intent multi-intent classifier that utilizes the output of IND task and detects the multiple intents without depending on the threshold. Extensive experiments demonstrate that our proposed model achieves superior results on two public multi-intent datasets.

1 Introduction

Natural language understanding (NLU) consists of two sub-tasks, including intent detection and slot filling which allow the dialogue system to create a semantic frame that summarizes the user's requests. Early works often approach these two tasks separately (Cortes and Vapnik, 1995; McCallum et al., 2000; Sarikaya et al., 2011; Yao et al., 2014; Vu, 2016). Considering intent detection and slot filling are highly related, recent works tend to model these two tasks jointly, where the correlation between the intent and slots are utilized (Goo et al., 2018; E et al., 2019; Qin et al., 2019; Zhou et al., 2021).

The works above only consider the scenario where each utterance has one intent. However, in real-life situations, users may express multiple intents in an utterance, thus making it difficult to apply single intent NLU models. Recently, several works have studied Multi-intent NLU problem. Gangadharaiah *et al.*(2019) investigated an attention-based neural network. Qin *et al.*(2020) proposed an Adaptive Graph Interactive Framework (AGIF). Qin *et al.*(2021) explored a non-autoregressive approach to speed up the inference time. However, these works all predict multiple intents with threshold, where the common practice is estimating label-instance probabilities and picking the intent labels whose probabilities are higher than the threshold value. We named them threshold-based models. The main issue of threshold-based models is that they are not robust to the threshold settings. As shown in Figure 1, the correct intents for the utterance are 'GetWeather' and 'SearchScreeningEvent'. Although the model can detect that 'GetWeather' and 'SearchScreeningEvent' are the two most probable intents, the threshold-based model only considers 'GetWeather' as the intent due to the threshold which is usually set as 0.5.

In this paper, we propose a transformer-based **T**hreshold-**f**ree **M**ulti-**N**LU model (TFMN) and detects multiple intents without relying on the threshold. Specifically, we leverage the upper layers of a transformer-based encoder to generate multi-grain representations. Next, we fully exploit the annotations from original multiple intents data and propose an Intent Number Detection (IND) task. The motivation is to allow the model to detect the intent



Utterance: Will it be freezing in the current position and also what is the showtime for arsho
Intent: 'GetWeather', 'SearchScreeningEvent'

Figure 1: A threshold-based multi-intent detection example in MixSNIPS with given utterance and intent labels. Threshold, which is the dash line, is set to 0.5.

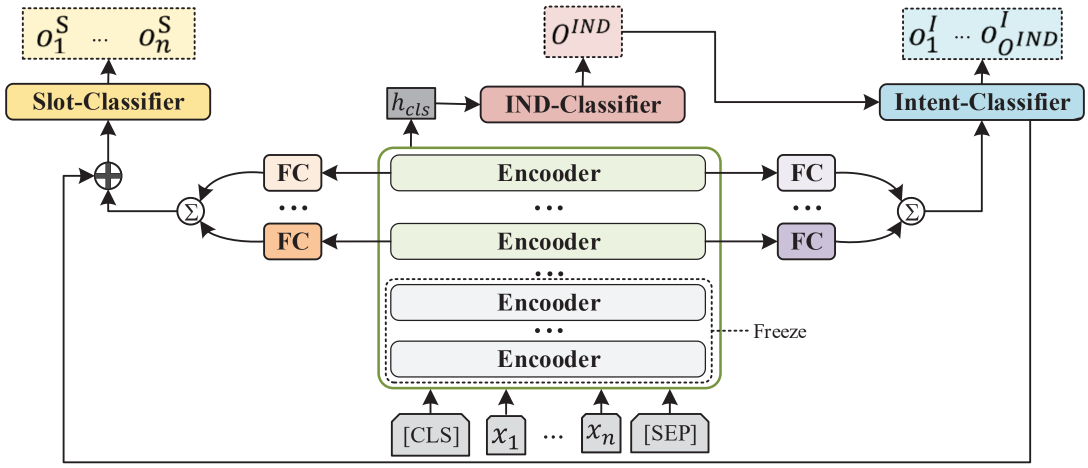


Figure 2: The architecture of transformer-based TFMN model.

numbers in a given utterance. Then we propose a threshold-free intent classifier that utilizes the output of IND task to detect the multiple intents.

We validate TFMN on two public datasets (Qin et al., 2020): MixATIS and MixSNIPS, and show that our method outperforms competitive baselines. The contributions of our work are summarized as follows: (1)We propose a novel threshold-free Multi-NLU model based on transformers.(2) We propose IND task, a feasible task to improve the multi-intent NLU without additional manual annotation, and a threshold-free multi-intent classifier that detects multiple intents without relying on threshold. (3) We present extensive experiments demonstrating the effectiveness of our approach.

2 Problem Formulation

Given an input sequence $X = (x_1, ..., x_n)$, multi-intent detection is defined as a multi-label classification task that outputs $O^I = (o_1^I, ..., o_m^I)$, where m is the number of predicted intent labels. Slot filling task can be regarded as a sequence labeling task that outputs a slot sequence $O^S = (o_1^S, ..., o_n^S)$.

3 Approach

In this section, we first introduce the architecture of TFMN model, then detail the proposed IND task and threshold-free intent classifier.

3.1 Threshold-free Multi-intent NLU Model

The architecture of our model is illustrated in Figure 2. TFMN includes a transformer-based encoder with L layers and three task-specific classifiers.

Multiple Intent Detection Following (Qin et al., 2019), we perform a token-level multiple intent detection which can be formalized as a sequence labeling problem that maps the input utterance $X = (x_1, ..., x_n)$ to sequence of intent label $O^I = (o_1^I, ..., o_n^I)$. According to (Jawahar et al., 2019; Rogers et al., 2020), transformer-based encoder tends to capture syntactic information in the middle and semantic information at the top layers. Therefore, we take the top j layers of the encoder to form multi-grain intent features. First, we map each hidden layer into a different feature space via a fully connected layer, then we combine hidden layers by adding them together:

$$h^I = \sum_{n=L-j}^{L} w_n^I h_n \quad (1)$$

where w_n^I are trainable parameters and h_n are different hidden layers. We then generate intent logits with the intent feature h_I:

$$l^I = w_i h^I \quad (2)$$

where w_i are trainable parameters. The intent logits will be used to provide token-level intent information for slot filling and detect the final multiple intent labels which we will detail in Section 3.3.

Slot Filling Similar to intent detection, We leverage the top j layers of a transformer-based encoder for slot filling. The slot features h^S are generated by combining hidden layers and concatenating with token-level intent information:

$$h^S_{temp} = \sum_{n=L-j}^{L} w_n^S h_n \quad (3)$$

$$h^S = h^s_{temp} \oplus l^I \quad (4)$$

then slot classifier computes the slot prediction:

$$p_t^S = softmax(w_a LeakyReLU(w_b h_t^S)) \quad (5)$$

where w_a and w_b are trainable parameters.

3.2 Intent Number Detection

To achieve threshold-free multi-intent detection, we propose an Intent Number Detection task which trains with the intent detection and slot filling in a multi-task fashion. In IND task, we fully utilize the original intent label annotations by calculating the numbers of intents in each utterance and forming the intent number labels Y^{IND}. Then we train the model to detect how many intents are there in the input utterance with Y^{IND}. Specifically, we take the output of [CLS] token from the last hidden layer h_{cls} as representation for IND task to classify:

$$p^{IND} = softmax(w_{ind}h_{cls}) \tag{6}$$

$$O^{IND} = argmax(p^{IND}) \tag{7}$$

We use cross-entropy to optimize IND task:

$$\mathcal{L}_{IND} = -\sum_{k} y_k^{IND} log p_k^{IND} \tag{8}$$

3.3 Threshold-free Intent Classifier

Once having the intent logits l^I and being able to predict the intent numbers with the proposed IND task, we send l^I into a sigmoid activation function:

$$p_t^I = sigmoid(l_t^I) \tag{9}$$

where p_t^I is the intent probability distribution of tth token in the utterance. Since the final output should be the utterance-level intent detection, we sum p_t^I up for utterance-level intent probability distribution P^I, and choose the top O^{IND}, which is the predicted intent number of the utterance, most probable intent label as the final result $O^I = (o_1^I, ..., o_{O^{IND}}^I)$.

3.4 Multi-Task Training

Our model optimizes the parameters jointly. Multiple intent detection is trained with binary cross-entropy and slot filling is trained with cross-entropy. The total loss of TFMN is the weighted sum of three losses:

$$\mathcal{L}_{total} = \alpha \cdot \mathcal{L}_{ID} + \beta \cdot \mathcal{L}_{SF} + \lambda \cdot \mathcal{L}_{IND} \tag{10}$$

with three hyper-parameters α, β, and λ to balance.

4 Experiments

4.1 Datasets

We conduct experiments on two public multi-intent NLU datasets[1]. They are MixATIS (Qin et al., 2020) collected from ATIS dataset (Hemphill et al., 1990) with 13162/759/828 utterances for train/validate/test and MixSNIPS (Qin et al., 2020) collected from SNIPS dataset (Coucke et al., 2018) with 39776/2198/2199 utterances for train/validate/test. Both of the datasets have the ratio of sentences with 1~3 intents as [0.3, 0.5, 0.2].

4.2 Experimental Settings

For TFMN, we use the English uncased Bert-Base model (Devlin et al., 2019) which consists of 12 hidden layers, 12 heads, and the hidden size is 768. For fine-tuning, we freeze the bottom half of Bert and empirically choose the top 4 layers[2] to generate representations. The batch size is 128 and the epoch is 80. Adam is used for optimization with learning rate of 2e-5. The hyper-parameters of loss are empirically set as α: β: λ= 0.6: 1: 1 for MixATIS and α: β: λ= 0.7: 0.9: 1 for MixSNIPS. We evaluate the performance of slot filling with F1 score, intent detection with accuracy, and the NLU semantic frame parsing with overall accuracy.

4.3 Baselines

We compare our model with both single-intent and multi-intent baselines. For single-intent models to handle multi-intent utterances, multiple intent labels are connected with "#" and treated as a single label, named as *concat* version. For multi-intent baselines, they are all threshold-based models, named as *thresh* version. We also obtain our own pre-trained language model (PLM) baselines for comparison which are Bert-baseline and Roberta-baseline. Following (Chen et al., 2019), we obtain the hidden state of the first special token ([CLS]) with sigmoid function for detecting multi-intent based on threshold and use hidden states of utterance tokens for slot filling.

4.4 Results

The main results are illustrated in Table 1. We observe that TFMN model outperforms previous state-of-the-art (SOTA) baselines significantly. On slot filling, our model outperforms GL-GIN 1.5% on MixSNIPS. For multiple intent detection, we achieve 3.5% and 2.1% improvement compared with GL-GIN on MIxATIS and MixSNIPS respectively. On overall accuracy, our model shows strong performance which surpasses GL-GIN 6.7% on

[1] https://github.com/LooperXX/AGIF

[2] More information about utilizing layers is provided in Appendix A.5.

Model	MixATIS			MixSNIPS		
	Slot (F1)	Intent (Acc)	Overall (Acc)	Slot (F1)	Intent (Acc)	Overall (Acc)
Bi-Model (*concat*) (2018)	83.9	70.3	34.4	90.7	95.6	63.4
SF-ID (*concat*) (2019)	87.4	66.2	34.9	90.6	95.0	59.9
Stack-Propagation (*thresh*) (2019)	87.8	72.1	40.1	94.2	96.0	72.9
Joint Multiple ID-SF (*thresh*) (2019)	84.6	73.4	36.1	90.6	95.1	62.9
AGIF (*thresh*)(2020)	86.7	74.4	40.8	94.2	95.1	74.2
GL-GIN (*thresh*)(2021)	**88.3**	76.3	43.5	94.9	95.6	75.4
Bert-baseline (*thresh*)	86.3	74.5	44.8	95.5	95.6	80.1
Roberta-baseline (*thresh*)	85.0	78.3	47.8	95.9	97.5	83.2
TFMN (Bert-base)	88.0	**79.8**	**50.2**	**96.4**	**97.7**	**84.7**

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

Model	MixATIS		
	Slot (F1)	Intent (Acc)	Overall (Acc)
TFMN	88.0	79.8	50.2
-w/o T-free Cls	87.1	77.3	47.0
-w/o T-free Cls & IND task	86.3	76.8	46.7

Table 2: Ablation study. *T-free Cls* indicates threshold-free intent classifier.

Model	MixATIS			
	Int-1	Int-2	Int-3	Avg.
AGIF	96.5	83.7	76.7	85.6
GL-GIN	96.5	94.6	87.5	92.8
Bert-baseline	94.4	87.8	83.5	92.2
Roberta-baseline	97.2	89.5	78.0	88.2
TFMN	**98.6**	**99.7**	**99.3**	**98.9**

Table 3: A comparison of intent number prediction between threshold-based and threshold-free approaches. The evaluation metric is accuracy. **Int-#** means the utterance with the number of "#" intent. Avg. is the average accuracy.

MixATIS and 9.3% on MixSNIPS. When comparing with PLM baselines, although Bert-baseline has much worse performance than Roberta-baseline, our Bert-base TFMN model still manages to outperform Roberta-baseline. The results suggest that our approach brings significant improvements to multi-intent NLU. We believe this is due to the proposed IND task which fully exploits original intent annotations and threshold-free intent classifier that allows our model to detect multiple intents without a threshold and lead to performance gains[3].

4.5 Ablation Study

We compare TFMN with two simplified versions, *-w/o T-free Cls* and *-w/o T-free Cls & IND task* in Table 2 to analyze the effectiveness of threshold-free intent classifier and IND task. We can see that as the threshold-free intent classifier is removed, the performances drop 0.9%, 2.5%, and 3.2% on slot F1, intent accuracy, and overall accuracy respectively. We attribute this to the fact that the threshold-free approach can better detect the intent number in an utterance compare to threshold strategy. We further remove the INP task and the performance again drops 0.8%, 0.5%, and 0.3% on slot F1, intent accuracy, and overall accuracy respectively. This indicates the effectiveness of introducing the INP task to multi-intent NLU.

[3]More comparisons about inference speed and PLMs are provided in Appendix A.1 A.2.

4.6 Threshold-based vs Threshold-free

To compare threshold-based and threshold-free approaches, we evaluate how well a model can detect the number of intents. The results are demonstrated in Table 3. We obtained that the threshold-free model, TFMN, significantly outperforms the threshold-based baselines. Our model achieves 2.1%, 5.1%, 11.8%, and 6.1% improvements on one to three intent utterances and average accuracy over the previous SOTA baseline, GL-GIN. We find it interesting that threshold-based models predict intent number well when there is one intent in the utterance and become worse as the intent number increase while TFMN shows more consistency.

5 Conclusion

In this paper, we propose TFMN model which detects intent numbers in an utterance by a novel IND task that does not require additional manual annotations. Then we propose a threshold-free intent classifier to detect multiple intents without relying on the threshold. Extensive experiments show that TFMN achieves performance gains over strong baselines, and verify the effectiveness of the proposed IND task and threshold-free intent classifier.

References

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

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning*.

A. Coucke, A. Saade, Adrien Ball, Théodore Bluche, A. Caulier, D. Leroy, Clément Doumouro, Thibault Gisselbrecht, F. Caltagirone, Thibaut Lavril, Maël Primet, and J. Dureau. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *ArXiv*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Haihong E, Peiqing Niu, Zhongfu Chen, and Meina Song. 2019. 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*, pages 5467–5471, Florence, Italy. Association for Computational Linguistics.

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, Volume 1 (Long and Short Papers)*, pages 564–569, Minneapolis, Minnesota. Association for Computational Linguistics.

Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. 2019. Representation degeneration problem in training natural language generation models. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

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 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

C. T. Hemphill, J. Godfrey, and G. Doddington. 1990. The atis spoken language systems pilot corpus. In *HLT*.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Andrew McCallum, Dayne Freitag, and Fernando CN Pereira. 2000. Maximum entropy markov models for information extraction and segmentation. In *Icml*.

Libo Qin, W. Che, Yangming Li, Haoyang Wen, and T. Liu. 2019. A stack-propagation framework with token-level intent detection for spoken language understanding. In *EMNLP/IJCNLP*.

Libo Qin, Fuxuan Wei, Tianbao Xie, Xiao Xu, Wanxiang Che, and Ting Liu. 2021. Gl-gin: Fast and accurate non-autoregressive model for joint multiple intent detection and slot filling.

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

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Ruhi Sarikaya, Geoffrey E Hinton, and Bhuvana Ramabhadran. 2011. Deep belief nets for natural language call-routing. In *ICASSP*. IEEE.

Ngoc Thang Vu. 2016. Sequential convolutional neural networks for slot filling in spoken language understanding. In *INTERSPEECH*.

Yu Wang, Yilin Shen, and Hongxia Jin. 2018. A bi-model based rnn semantic frame parsing model for intent detection and slot filling. In *NAACL*.

Kaisheng Yao, Baolin Peng, Yu Zhang, Dong Yu, Geoffrey Zweig, and Yangyang Shi. 2014. Spoken language understanding using long short-term memory neural networks. In *2014 IEEE Spoken Language Technology Workshop (SLT)*. IEEE.

Peilin Zhou, Zhiqi Huang, Fenglin Liu, and Yuexian Zou. 2021. Pin: A novel parallel interactive network for spoken language understanding. *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 2950–2957.

A Appendix

Model	MixATIS	
	Latency (s)	Speedup
Stack-Propagation	34.5	8.2x
Joint Multiple ID-SF	45.3	11.0x
AGIF	48.5	11.8x
GL-GIN	4.2	1.0x
TFMN	4.1	1.0x

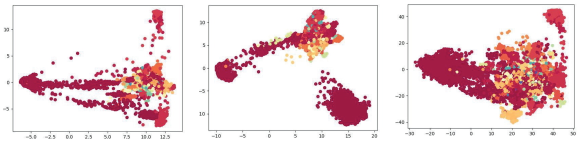
Table 4: Inference speed comparison. The speed is evaluated by running an epoch on the MixATIS dataset with batchsize set to 32 for each model.

A.1 Speed

Most previous works approach slot filling with autoregressive models, which lead to slow inference speed because they are not parallelizable. GL-GIN (Qin et al., 2021) is the most recent study that propose a non-autoregressive model with graph interaction layer. Our model also has a non-autoregressive paradigm, so we follow the speed test in (Qin et al., 2021) by running the model on the MixATIS test data by fixing batchsize to 32 for one epoch. The results are shown in Table 4. As indicated, our model ends up having similar inference speed as GL-GIN model, but with significant performance gains. When comparing with auto-regressive models, our model achieves x8.2, x11.0, x11.8 speedup compared with stack-propogation , Joint Multiple ID-SF, and AGIF.

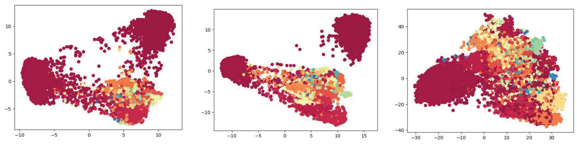
A.2 Visualization

In Table 1, we notice an interesting outcome when comparing the PLM-based baselines with previous state-of-the-art baselines. PLM-based baselines tend to have better performance in intent detection, which is utterance-level, and come short in slot filling, which is token-level, especially on MixATIS dataset. We argue that this is due to the representation degeneration problem (Gao et al., 2019), which is that the output embedding space is squeezed into a narrow cone. And such anisotropic shape limits the expressiveness of word embedding. On the other hand, our proposed model, which is also PLM-based, shows strong performances. So we visualize the token representation of bert-baseline, roberta-baseline, and TFMN from two datasets in Figure 3 and Figure 4. As shown, the visualizations of bert-baseline and roberta-baseline from two dataset are anisotropic and sparse while the visualizations of TFMN are much more expressive. This results explain the insights we mention above and also show that our approach of leveraging multiple layers of Bert model for slot features alleviates the representation degeneration problem.



(a) Bert (b) Roberta (c) TFMN

Figure 3: PCA visualization of the slot representations on MixATIS dataset.



(a) Bert (b) Roberta (c) TFMN

Figure 4: PCA visualization of the slot representations on MixSNIPS dataset.

Hyper-parameter	Search Range
Learning Rate	{2e-5, 5e-5}
Loss weight of intent classifier	{0.6, 0.7, 0.8, 1}
Loss weight of slot classifier	{0.9, 1, 1.2}
Loss weight of IND classifier	{0.9, 1}

Table 5: Hyper-parameter search range of our proposed TFMN model.

A.3 Computing Infrastructure and Computation Time

All experiments are conducted using a single Geforce RTX 3090 GPU. When training for 80 epochs, time costs approximately 42 minutes on MixATIS and 105 minutes on MixSNIPS.

A.4 Number of Parameters

TFMN includes one Bert encoder and three task-specific classifiers. The parameters in TFMN are slightly larger than uncased Bert-base model which is around 111 million.

A.5 Hyper-parameter and Bert Layers

The main hyper-parameters of TFMN are the learning rate and weights of losses for Multiple intent detection, slot filling, and intent number detection. We randomized search for the best setting to maxi-

Bert layers	MixATIS		
	Slot (F1)	Intent (Acc)	Overall (Acc)
{10, 11, 12}	87.4	79.1	48.7
{8, 9, 10, 11, 12}	85.6	79.4	48.7
{9, 10, 11, 12}	88.0	79.8	50.2

Table 6: Results of different combinations of bert layers for representation.

mize the semantic frame accuracy. Detailed search range of hyper-parameters are given in Table 5.

Empirically, we choose the top 4 layers of Bert for generating our representations. We have also tried out different combinations as shown in Table 6. We argue that only choosing the top 3 layers of Bert does not offer enough linguistic information while choosing the top 5 layers will bring in noise which leads to a performance decrease.

A.6 Dataset Explanation

We conduct our experiments on two English Multi-intent NLU datasets. They are the cleaned version of MixATIS and MixSNIPS, because they found some repeated sentences in the original MixATIS and MixSNIPS datasets so that they clean these two datasets and recommend using the cleaned version.