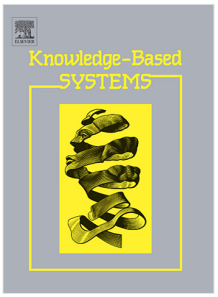
Journal Pre-proof



AISE: Attending to Intent and Slots Explicitly for better spoken language understanding

Peng Yang, Dong Ji, Chengming Ai, Bing Li

PII: S0950-7051(20)30666-3
DOI: https://doi.org/10.1016/j.knosys.2020.106537
Reference: KNOSYS 106537

To appear in: *Knowledge-Based Systems*

Received date : 16 April 2020
Revised date : 13 October 2020
Accepted date : 13 October 2020

Please cite this article as: P. Yang, D. Ji, C. Ai et al., AISE: Attending to Intent and Slots Explicitly for better spoken language understanding, *Knowledge-Based Systems* (2020), doi: https://doi.org/10.1016/j.knosys.2020.106537.

AISE: Attending to Intent and Slots Explicitly for Better Spoken Language Understanding

Peng Yang[a,b,c,*], Dong Ji[a,c], Chengming Ai[a,c], Bing Li[a,c]

[a]*School of Computer Science and Engineering, Southeast University, Nanjing, China*
[b]*School of Cyber Science and Engineering, Southeast University, Nanjing, China*
[c]*Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education, Nanjing, China*

Abstract

Spoken language understanding (SLU) plays a central role in dialog systems and typically involves two tasks: intent detection and slot filling. Existing joint models improve the performance by introducing richer words, intents and slots semantic features. However, methods that model the explicit interactions between these features have not been further explored. In this paper, we propose a novel joint model based on the position-aware multi-head masked attention mechanism, which explicitly models the interaction between the word encoding feature and the intent-slot features, thereby generating the context features that contribute to slot filling. In addition, we adopt the multi-head attention mechanism to summarize the utterance-level semantic knowledge for intent detection. Experiments show that our model achieves state-of-the-art results and improves the sentence-level semantic frame accuracy, with 2.30% and 0.69% improvement relative to the previous best model on the SNIPS and ATIS datasets, respectively.

Keywords: Spoken language understanding, Intent detection, Slot filling, Position-aware multi-head masked attention mechanism

*Corresponding author
Email address: pengyang@seu.edu.cn (Peng Yang)

1. Introduction

In recent years, dialog systems have developed rapidly and are now widely used in various human-computer interaction scenarios such as voice assistants and robots. Spoken language understanding (SLU) is a key technology for building such systems and typically involves two tasks: intent detection and slot filling[1, 2]. Intent detection determines what the user wants to do, while slot filling extracts semantic concepts related to the users intent. As shown in Figure 1, given an utterance, an SLU system will annotate a specific intent for the whole utterance and different slot labels for each word in the utterance.

Utterance	find	on	dress	parade
	↓	↓	↓	↓
Gold Slot	O	B-movie_name	I-movie_name	I-movie_name
Gold Intent	SearchScreeningEvent			

Fig. 1 An example with intent and slot annotation (BIO format), which indicates the slot of the movie name from an utterance with an intent SearchScreeningEvent.

As the semantic concepts of an utterance, intent and slot are supposed to share the semantic features and rely on each other. To take advantage of the semantic correlation between the two tasks, multi-task joint models for SLU have been widely proposed. Zhang[3] and Hakkani-Tur[4] utilized the GRU-based and LSTM-based RNN to capture the shared word semantic features for intent detection and slot filling, respectively. These methods can be characterized by shared parameters, but the explicit relationship between the intent and slots is not established. Since the slots usually depend strongly on the intent, Goo[5] and Li[6] proposed different gate mechanisms to incorporate the intent information for slot filling. However, simple gate mechanisms can be risky [7], and none of the works reported to date model the slot label dependency. Zhu[8] proposed a focus mechanism that improved the attention mechanism with exact alignment to model the slot label dependency, but their method does not consider intent

detection. Liu[9] utilized an RNN to model the slot label dependency. Recently, Qin[10] proposed to perform token-level intent detection, followed by feeding the inferred intent feature to the aligned LSTM-based model for slot filling, and achieved superior performance. Although the intent and slot features are explicitly considered during slot filling, this method only models the interactions between the input features implicitly by LSTM.

Previous works have sought to capture more richer word, intent and slot semantic features for better performance. Slot filling is essentially a sequence labeling task, and there is a dependency between the slot labels. In addition, the intent information contributes to slot filling. Therefore, it is useful to incorporate the intent and slot information. However, the explicit interactions between these features have not been modelled effectively and may generate beneficial features more efficiently and make the learning process more interpretable.

In this paper, we propose a novel joint model for joint intent detection and slot filling by introducing the position-aware multi-head masked attention (PMMAtt) mechanism, where the explicit feature interactions are modelled as the inner product of the word encoding vector and intent-slot feature vectors. These interactions are further normalized into weights, indicating how much the model should pay attention to the corresponding feature. Moreover, the multi-head attention mechanism is adopted to capture sentence-level semantic information in order to improve the robustness of intent detection. Since the proposed joint model **A**ttends to **I**ntent and **S**lots **E**xplicitly, we name it **AISE**.

To summarize, the key contributions of this paper are as follows:

1. We introduce the PMMAtt mechanism to model explicit feature interactions between multiple tasks, which can guide models to rationally improve the performance and enable more efficient and interpretable learning.
2. We conduct extensive experiments, and the results on two publicly available datasets demonstrate the effectiveness of the proposed model.
3. We analyse the effect of different model components on the overall performance. The visualization of the learned weights illustrates the interpretability of our

model.

2. Related work

2.1. Intent detection

Generally, intent detection is treated as the sentence-level classification task, and various classification models such as SVM[11] and Adaboost[12] can be applied. As a text classification task, the performance of intent detection depends on the order of each word in the sentence. Traditional machine learning methods find it difficult to capture sequence information, limiting the model performance. With the development of deep learning, models based on deep learning such as the RNN[13], CNN[14], self-attention[15] and capsule network[16, 17] have surpassed the methods based on traditional machine learning in text classification tasks[18]. In [19], enriched word embedding is used as input for the bi-directional LSTM to carry out intent detection. Recently, Raymond[20] used the capsule network for intent detection and achieved excellent results. The method that we proposed for intent detection is different from the works mentioned above. We propose a novel joint model to model explicit feature interactions between multiple tasks.

2.2. Slot filling

In the current method, the slot filling task is converted into a sequence labeling task. The traditional method uses factorized probabilistic models such as HMM, MEMM and CRF to solve this problem. Some deep learning methods are also used for sequence labeling tasks, and their performance is better than that of the traditional machine learning methods. Recently, Tan[21] introduced the self-attention mechanism to perform sequence labeling tasks. Due to the limited training data in SLU, transfer learning techniques such as Fasttext[22], Elmo[23] and BERT[24] are also widely used. The pre-trained language model BERT was employed for intent detection and slot filling and achieved state-of-the-art result in [25]. The difference between these methods and our method is

that we introduced the PMMAtt mechanism to learn explicit feature interactions between words, intents and slots.

2.3. Joint tasks

In the early research, pipelined approaches that perform intent detection followed by slot filling were adopted in SLU. However, these approaches may lead to error propagation and redundancy of the model in the pipelined manner. In recent years, multi-task joint learning methods have become the main research direction of SLU.

Implicit Joint model This kind of model implicitly learns the correlation between intent detection and slot filling tasks by sharing parameters. Zhang[3] used the GRU as a shared encoder to capture semantic features for intent detection and slot filling. Similarly, Hakkani-Tur[4] adopted the LSTM for modeling the two tasks jointly. Liu and Lane[9] proposed an aligned RNN joint model with the attention mechanism that provides additional information that contributes to the tasks. Since the correlations between two tasks are utilized implicitly by optimizing the joint loss, these works lack interpretability and can be further improved.

Intent-Augmented model This type of method explicitly feeds intent information to the slot filling model. Goo[5] proposed a slot-gated mechanism that focuses on learning the relationship between the intent and slot attention vectors. Li[6] proposed an intent-augmented gating mechanism to fully utilize the semantic correlation between slot and intent. Qin[10] performed the token-level intent detection to improve the robustness of intent detection and proposed a stack-propagation framework that incorporates intent information to guide the slot filling. Similar to these methods, we also feed the intent information to our slot decoder. However, unlike our method, the previously developed methods do not explicitly model the interaction between the input features from the model perspective. By contrast, our slot decoder utilizes the PMMAtt mechanism to solve the problem effectively, making the model more interpretable and further improving the performance.

Bi-directional Interrelated model This method aims to model the bi-directional interrelationship between the intent and slot. Zhang[26] proposed a capsule network with a dynamic re-routing mechanism that models the hierarchical relationship between words, slots, and intents. Haihong[27] designed a new iterative mechanism to enhance the bi-directional interconnection between intent and slot. These models usually require the design of an iterative mechanism, making the model more complicated. Our model structure is relatively simple and easy to understand. Moreover, with the attention mechanism, the proposed model can be trained in parallel.

3. Approach

In this section, we present the **AISE** model for the SLU task. The model architecture is illustrated in Fig. 2 and consists of three main components: 1) a shared word encoder; 2) a multi-head attention-based sentence-level intent detector; and 3) a PMMAtt-based slot decoder.

3.1. Shared word encoder

Given an input utterance $u = (\mathrm{w}_1, \mathbf{w}_2, ..., \mathbf{w}_T)$ of T words where $\mathbf{w}_t \in \mathbf{R}^{D_W}$ is a word embedding. In joint models, intent detection shares the same word encoder with slot filling to learn the contextual semantic encoding of each word. Neural network architectures such as the RNN[13], CNN[14], and self-attention[15] are commonly applied to build such encoders. In this work, we adopt the bidirectional LSTM (Bi-LSTM) [28] as the shared encoder.

$$\begin{aligned} \overrightarrow{\mathbf{h}_t} &= \overrightarrow{LSTM}(\mathbf{w}_t, \overrightarrow{\mathbf{h}_{t-1}}), \\ \overleftarrow{\mathbf{h}_t} &= \overleftarrow{LSTM}(\mathbf{w}_t, \overleftarrow{\mathbf{h}_{t+1}}). \end{aligned} \tag{1}$$

The Bi-LSTM generates the forward hidden states $\overrightarrow{\mathbf{h_t}}$ and backward hidden states $\overleftarrow{\mathbf{h}_t}$ at each time step t. Then, $\overrightarrow{\mathbf{h_t}}$ and $\overleftarrow{\mathbf{h}_t}$ are concatenated to obtain the hidden states $\mathbf{h}_t = [\overrightarrow{\mathbf{h}_t}, \overleftarrow{\mathbf{h}_t}]$ at time step t, where $\mathbf{h}_t \in \mathbf{R}^{D_E}$. The final output sequence of the shared encoder is $\mathbf{H} = (\mathbf{h}_1, ..., \mathbf{h}_T)$.

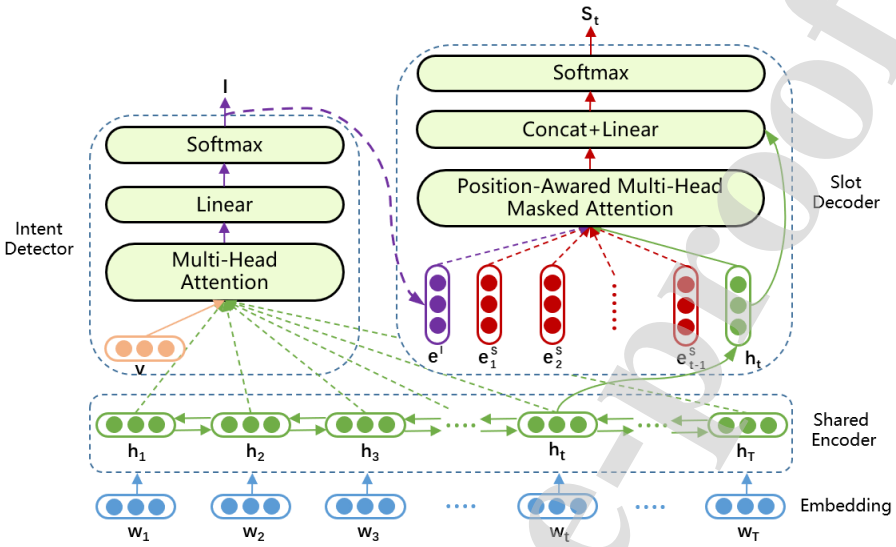


Fig. 2 Illustration of our AISE model for intent detection and slot filling. The model consists of a shared encoder, intent detector and slot decoder. $\mathbf{e}^I$ and $\mathbf{e}^S_{t-1}$ denote the embeddings of intent and slot, respectively, at position $t-1$, which are the context inputs of the slot decoder.

3.2. Intent detector

Intent detection is usually modeled as a sentence-level classification problem, which receives the encoding sequence $\mathbf{H} = (\mathbf{h}_1, ..., \mathbf{h}_T)$ output from the shared word encoder. Intent detector aims to aggregate the variable-size encoding sequence $\mathbf{H}$ into a fixed-size intent context vector $\boldsymbol{c}^{\boldsymbol{I}}$ that contributes to the task.

The common aggregation methods include the following: 1) last pooling; 2) max pooling; 3) average pooling; 4) capsule pooling[16, 17, 29]; and 5) attention pooling[30]. The attention pooling can be regarded as an extension of average pooling, which can dynamically adjust the weight of each vector. In our intent detector, multi-head attention pooling is adopted, as depicted in Fig. 2.

The intent context feature $\mathbf{c}^I$ is formulated as:

$$\boldsymbol{\alpha}^m = softmax(\frac{(\mathbf{q}\mathbf{W}_q^{I,m})^T \mathbf{W}_k^{I,m}\mathbf{H}}{\sqrt{d_q}}), \tag{2}$$

$$\boldsymbol{c}^{I,m} = \boldsymbol{\alpha}^{m}\mathbf{W}_{v}^{I,m}\mathbf{H}, \tag{3}$$

$$\boldsymbol{c}^{I} = \mathbf{W}^{IO}[\boldsymbol{c}^{I,1}, ..., \boldsymbol{c}^{I,N_I^H}], \tag{4}$$

where N_I^H denotes the number of the attention head, $\mathbf{W}_q^{I,m}$, $\mathbf{W}_k^{I,m}$ and $\mathbf{W}_v^{I,m}$ are Query, Key and Value parameter matrices of the m-th head, respectively. $\mathbf{q} \in \mathbf{R}^{D_Q}$ is a trainable query vector, [...] is the concatenation operation and W^{IO} is the output parameter matrix of attention pooling.

Then, $\mathbf{c}^I$ is utilized for intent detection:

$$\mathbf{p}^I = softmax(\mathbf{W}^I\mathbf{c}^I + \mathbf{b}^I), \tag{5}$$

$$y^I = argmax(\mathbf{p}^I), \tag{6}$$

where p^I is the intent label distribution, and y^I is the output intent label.

3.3. PMMAtt-based Slot Decoder

In this work, we introduce the position-aware multi-head masked attention (PMMAtt) mechanism to design a novel slot decoder, as shown in Fig. 2. At each position t, it has three inputs: 1) the word encoding $\mathbf{h}_t$; 2) the intent embedding $\mathbf{e}^I$; and 3) the left side slot embedding sequence $\mathbf{E}^S = (\mathbf{e}_1^S, ..., \mathbf{e}_{t-1}^S)$. The slot decoder sequentially predicts the slot label at each position from left to right.

The PMMAtt mechanism consists of two parts: multi-head masked attention[15] and relative positional encodings[31, 32]. Here, we adopt the method in [32] that injects the relative positional information to attention scores. In this case, the two parts are also called content-based attention and position-based attention.

In the slot decoder, we take the word encoding $\mathbf{h}_t$ as the Query input and the concatenation of $\mathbf{e}^I$ and $\mathbf{E}^S$ as the inputs of Key and Value. Then, the attention scores under each head are calculated:

$$\mathbf{E} = [\mathbf{e}^I, \mathbf{E}^S], \tag{7}$$

$$\begin{aligned}\mathbf{A}_{t,j}^{rel,n} = (\mathbf{W}_q^{S,n}\mathbf{h}_t)^T\mathbf{W}_{k,E}^{S,n}\mathbf{E}_j + u^T\mathbf{W}_{k,E}^{S,n}\mathbf{E}_j \\ +(\mathbf{W}_q^{S,n}\mathbf{h}_t)^T\mathbf{W}_{k,R}^{S,n}R_{t-j} + v^T\mathbf{W}_{k,R}^{S,n}\mathbf{R}_{t-j},\end{aligned} \tag{8}$$

where $\mathbf{W}_q^{S,n}, \mathbf{W}_{k,E}^{S,n}, \mathbf{W}_{k,R}^{S,n}$ are the Query, content-based and position-based Key parameter matrix of the ut$n$-th head respectively. $\mathbf{A}_{t,j}^{rel,n}$ is the unnormalized PMMAtt weight of $\mathbf{E}_j$ at position t. $\mathbf{R}_{t-j}$ is the positional encoding with relative distance $t-j$, which is a sine function[15]:

$$\mathbf{R}_{t-j} = sin((t-j)/10000^{2i/D_P}) \tag{9}$$

where D_P is the dimension of position encoding.

Note that $\mathbf{E}_0 = \mathbf{e}^I$ is the intent embedding, which has no positional information, so we set its positional attention score to 0, making the slot decoder only consider the content-based attention of the intent feature.

Then, we normalize the scores to obtain the PMMAtt weights $\boldsymbol{\beta}_t^n$:

$$\boldsymbol{\beta}_t^n = softmax(\boldsymbol{A}_t^{rel,n}) \tag{10}$$

A larger $\boldsymbol{\beta}_{t,j}^n$ indicates that $\mathbf{E}_j$ is more reliable for determining the inferred slot. In other words, the predicted slot has a stronger correlation with the intent or slots at some positions. $\boldsymbol{\beta}_t^n$ represents a feature selection strategy that can generate different aspects of the features.

Under each head, we compute the single-head slot context vector $\mathbf{c}^{S,n}$ as a weighted sum of the intent and slots features. The final slot context vector $\mathbf{c}^S$ is calculated by linearly transforming the concatenation of multiple single-head slot context vectors.

$$\mathbf{c}^{S,n} = \sum_{j=0}^{t-1}\boldsymbol{\beta}_{t,j}^n(\mathbf{W}_V^{S,n}\mathbf{E}_j) \tag{11}$$

$$\mathbf{c}^S = \mathbf{W}^{SO}[\mathbf{c}^{S,1}, ..., \mathbf{c}^{S,N_S^H}] \tag{12}$$

where $\mathbf{W}_V^{S,n}$ is the Value parameter matrix of the n-th head in the slot decoder, $\mathbf{W}^{SO}$ is the trainable parameter, and N_S^H is the number of head.

The word encoding $\mathbf{h}_t$ and slot context vector $\mathbf{c}^S$ are utilized for slot filling:

$$y_t^S = softmax(\mathbf{W}^S[\mathbf{c}^S, \mathbf{h}_t] + \mathbf{b}^S) \tag{13}$$

where y_t^S is the slot label of the t-th word, $\mathbf{W}^S$ is the weight matrix, and $\mathbf{b}^S$ is the bias.

3.4. Joint Optimization

Cross-entropy loss is used for both the intent detection and slot filling tasks. Specifically, the loss function of intent detection is:

$$L_I = -\sum_k \hat{y}_k^I log P(y_k^I) \tag{14}$$

The cross entropy loss for slot filling is:

$$L_S = -\sum_t \sum_k \hat{y}_{t,k}^S log P(y_{t,k}^S) \tag{15}$$

The model is optimized jointly, and the final objective function is defined as the weighted sum of the two loss functions:

$$L = (1 - w_{slot}) * L_I + w_{slot} * L_S \tag{16}$$

where $w_{slot} \in [0, 1]$ is a hyper-parameter that weighs the importance of slot filling.

4. Experiments

4.1. Datasets and Metrics

We conduct experiments on two publicly available datasets, namely, SNIPS[33] and ATIS[34], which are widely used in SLU research. Both datasets used in our paper follow the same format and partition as in [5]. The statistical information for the two datasets is presented in Table 1.

Referring to previous works, we evaluate the proposed model regarding intent detection using accuracy, slot filling using F1 score, and overall performance

Table 1 Dataset statistics.

Dataset	**SNIPS**	**ATIS**
Vocabulary Size	11,241	722
Average Sentence Length	9.05	11.28
#Intents	7	21
#Slot labels	72	120
#Training Set Size	13,084	4,478
#Development Set Size	700	500
#Test Set Size	700	893

using sentence level semantic frame accuracy, which is the fraction of sentences with correct intent and slots.

The slot F1 score is calculated as the harmonic average of the precision and recall of the slot chunk, which is composed of slots with the same type in one or more consecutive positions. In each slot chunk, the tag of the first slot must be B, and the tags of all other slots must be I.

Assuming that the number of gold slot chunks in the test set is N, the number of predicted slot chunks is M, and the number of correctly predicted slot chunks is K, the slot precision P is calculated as:

$$P = \frac{K}{M} \tag{17}$$

The slot recall R is calculated as:

$$R = \frac{K}{N} \tag{18}$$

Finally, the slot F1 score$F1$ is calculated as:

$$F1 = \frac{2 * P * R}{P + R} \tag{19}$$

4.2. Baselines

We compare the proposed model with the following existing models: 1) *Joint Seq* [4]; 2) *Atten.-Based* [9]; 3) *Slot-Gated Atten* [5]; 4) *Self-Attentive Model* [6]; 5) *Capsule-NLU* [26]; 6) *Bi-Model* [35]; and 7) *Stack-Propagation* [10]. These baselines can be classified into three categories: 1) and 2) are the implicit joint

models; 3), 4) and 7) are the intent-augmented joint models; and 5) and 6) are the bi-directional interrelated models.

For the *Joint Seq* and *Atten.-Based* models, we adopt the results reported by [5]. For the *Self-Attentive Model* and *Bi-Model*, we adopt the results reported by [10]. All results are obtained on the same two datasets.

4.3. Implementation details

Table 2 summarizes the hyper-parameters used in our model. The hyper-parameters are tuned on the development set.

Table 2 Summary of the hyper-parameters

	w_{slot}	*drop*	N^{lstm}	N_I^H	N_S^H	$D_{W\|E\|Q}$	D_P	$D_{I\|S}$
range	[0,1]	[0,1]	**N**+	**N**+	**N**+	**N**+	**N**+	**N**+
value	0.7	0.5	2	3	3	256	32	64

[1] **N**+ denotes the positive integer set.

[2] N^{lstm} is the number of BiLSTM layers; *drop* is the dropout rate;

[3] N_I^H,N_S^H are the numbers of heads in the intent detector and slot decoder, respectively;

[4] D_W, D_P, D_I, and D_S are the dimensions of the word, relative position, intent and slot embedding, respectively; D_E is the output dimension of the shared encoder; and D_Q is the dimension of query vector in the intent detector.

The trainable parameters including word embeddings are randomly initialized, and we use the Adam optimizer [36] to minimize the loss.

4.4. Main Results

We choose the model that performs best on the development set and then evaluate the results on the test set. Table 3 compares the experimental results of the proposed model with the reported results from baselines. An examination of the data presented in the table shows that our model achieves state-of-the-art results and outperforms the previous best results, with 1.0% improvement on the F1 score, 0.7% improvement on the intent accuracy, and 2.0% improvement on the overall accuracy in SNIPS dataset. In another ATIS dataset, we achieve

Table 3 Slot filling and intent detection results on two datasets (%).

Model	SNIPS			ATIS		
	Slot(F1)	Intent(Acc)	Overall(Acc)	Slot(F1)	Intent(Acc)	Overall(Acc)
Joint Seq[4]	87.3	96.9	73.2	94.3	92.6	80.7
Atten.-Based[9]	87.8	96.7	74.1	94.2	91.1	78.9
Slot-Gated[5]	88.8	97.0	75.5	94.8	93.6	82.2
Self-Attentive Model[6]	90.0	97.5	81.0	95.1	96.8	82.2
CAPSULE-NLU[26]	91.8	97.3	80.9	95.2	95.0	83.4
Bi-Model[35]	93.5	97.2	83.8	95.5	96.4	85.7
Stack-Propa.[10]	94.2	98.0	86.9	95.9	96.9	86.5
Our model(AISE)	**95.2**	**98.7**	**88.9**	**96.0**	**97.0**	**87.1**
Oracle(Intent)	95.5	-	-	96.1	-	-

[1] The improvement of our model over all baselines is statistically significant with $p < 0.05$ under t-test.

a 0.6% improvement on the overall accuracy. These results demonstrate the effectiveness of the proposed model.

In particular, our model improves sentence-level semantic frame accuracy by a large margin, where the relative improvement is approximately 2.3% and 0.69% for SNIPS and ATIS, respectively. We attribute the improvement to the following reasons: 1) We feed the embeddings of the inferred intent and slot labels to the slot decoder. Based on this prior knowledge, the slot decoder is more robust, and the conflict between intent detection and slot filling is alleviated; 2) The proposed model can select effective features for slot filling with the PMMAtt mechanism through explicit interactions, further improving the overall performance.

In addition, to evaluate the effect of intent information, we only train the slot filling model and feed the gold intent label to it in the training and inferring phase. As shown in the *oracle* row in Table 3, we find that the slot filling performance is further improved, which also demonstrates that intent information is beneficial for slot filling.

5. Analysis

In this paper, we decompose the joint model into three major components, each of which can be optimized separately. For the shared encoder, transfer

learning techniques can be applied instead of training it from scratch. For the intent detector, different pooling operations can be adopted. To further illustrate the effectiveness of the proposed slot decoder, we conduct experiments with BERT and different pooling operations. In addition, the learning results of the slot decoder is visualized to demonstrate its interpretability.

5.1. Effect of Joint Learning

The proposed model jointly learns the implicit dependency between intent and slots by sharing parameters. To verify the effectiveness of the joint learning, we conduct the experiments with following ablation:

- w/o joint learning, where the intent detection and slot filling have their own word embedding and encoder, respectively, and the intent label embedding fed to the slot decoder is randomly initialized. The other components remain unchanged. We call this model independent learning.

We choose the model that performs best on the development set in terms of the specific metric, then calculate the metric on the test set, and finally compute the overall accuracy of two independent optimal models on the test set.

Table 4 Ablation comparison of our proposed model on two datasets (%).

Model	**SNIPS**			**ATIS**		
	Slot(F1)	Intent(Acc)	Overall(Acc)	Slot(F1)	Intent(Acc)	Overall(Acc)
w/o joint learning	94.7	98.0	86.1	95.6	97.4	86.8
w/o intent feature	95.0	97.7	88.3	95.6	97.2	86.6
w/o relative position	94.8	98.1	87.7	95.9	97.4	87.0
w/o PMMAtt mechanism	93.9	98.2	86.7	95.3	96.5	86.2
Our model	**95.2**	**98.7**	**88.9**	**96.0**	97.0	**87.1**

The *w/o joint learning* row in table 4 shows the experimental results for the above model. It is observed that the intent detection and slot filling models perform well without shared parameters, but the overall performance decreases on both datasets. The performance degradation on SNIPS is significant. Although the intent accuracy increases on the ATIS, the overall accuracy still decreases. This result can be explained based on two effects: (1) The related tasks can

promote each other through joint learning; and (2) The consistency between the intent and slots of each utterance can be improved by joint learning, which models the relationship between intent and slots by sharing parameters, thereby enhancing the overall performance.

5.2. Effect of Slot Decoder

The proposed slot decoder takes the intent label as an additional input feature and adopts the PMMAtt mechanism to model the explicit interaction between the word encoding feature and intent-slot features. To evaluate the effectiveness of each feature or component used in the slot decoder, we conduct the experiments with the following ablations:

- w/o the intent feature, where the slot decoder does not rely on the output of the intent detector, and the intent embedding fed to the slot decoder is initialized randomly.
- w/o the relative position, where the relative position is removed from the PMMAtt mechanism, which degenerates into a multi-head attention mechanism.
- w/o the PMMAtt mechanism, where the intent feature, relative position and multi-head attention mechanism are removed, and the aligned hidden states of BiLSTM are used to predict each slot.

Table 4 shows the performance of our joint model on SNIPS and ATIS obtained after removing one or more components or features at a time. It is observed that each component or feature contributes to the overall performance.

If we remove the intent feature, we find that the slot F1 and overall accuracy decrease on both datasets. It is beneficial for slot filling and full model to explicitly model the relationships between intent and slot, which is consistent with the findings of the previous studies.

If we remove the relative position, the PMMAtt mechanism will degenerate into a multi-head attention mechanism. As shown in Table 4, the slot F1 and

overall accuracy drop by 0.4% and 1.2% on SNIPS and decrease slightly on ATIS. These results show that the relative position has a positive effect.

If we remove the PMMAtt mechanism, we see from Table 4 that the performance on both datasets decreases dramatically. The results further demonstrate the effectiveness of the proposed slot decoder, which can better capture the explicit interactions between word and intent-slots to guide the slot filling and improve the overall performance.

5.3. Effect of Intent Detector

In this section, we study the effect of the proposed intent detector with the following aggregation methods:

- Last. The last hidden states of the shared encoder are used as the intent context vector $\boldsymbol{c^I}$.
- Max. Max pooling is applied on the output hidden states of the shared encoder to generate the intent context vector $\boldsymbol{c^I}$.
- Average. Average pooling is applied on the output hidden states of the shared encoder to generate the intent context vector $\boldsymbol{c^I}$.
- Capsule. A capsule network with the dynamic routing-by-agreement described in Gong[29] is utilized to aggregate the encoding sequence output by the shared encoder. Since the output of the capsule network is a set of vectors called capsules, we map these capsules into the intent context vector $\boldsymbol{c^I}$ by a linear layer.

Table 5 shows the results of our model with different aggregation methods. We find that all of the variants of the intent decoder perform well, and the multi-head attention mechanism achieves the best performance among all variants in terms of the three metrics. The results further demonstrate the effectiveness of the proposed intent detector.

Table 5 Model performance with different aggregation methods (%).

Aggregator	SNIPS			ATIS		
	Slot(F1)	Intent(Acc)	Overall(Acc)	Slot(F1)	Intent(Acc)	Overall(Acc)
Last	95.0	97.9	88.6	95.7	96.2	85.8
Max	94.9	98.0	88.6	95.7	96.4	86.7
Average	94.9	98.1	88.6	95.7	96.5	86.5
Capsule	94.9	98.6	88.3	95.9	96.1	85.9
Multi-head Att.	**95.2**	**98.7**	**88.9**	**96.0**	**97.0**	**87.1**

5.4. Effect of Word Encoder

The word encoder is designed to model rich semantic features, which are critical for the performance of intent detection and slot filling. Similar to the baseline models in Section 4.2, the word encoder in our model is based on Bi-LSTM. Intuitively, a more powerful word encoder should be able to further improve the SLU performance. Recently, pre-trained language models with a fine-tuning approach have greatly improved the performance of various NLP tasks. In SLU, Chen[25] built a joint model based on BERT[24], which simply feeds the output of BERT to a softmax layer and significantly outperforms the traditional models. Qin[10] fed the output of BERT to the stack-propagation framework, further improving the model performance. Table 6 shows the performance of the above two models.

To verify the effectiveness of our slot decoder under the pre-trained model, we replace BiLSTM by BERT-base with a fine-tuning approach. Specifically, we feed the hidden states of the special token [CLS] to a softmax layer for intent detection. For slot filling, the output of BERT is fed to our slot decoder. Since BERT adopts wordpiece to tokenize each input token into multiple sub-tokens, we only use the hidden states corresponding to the first sub-token as the input to the slot decoder. Finally, the structure of our model based on BERT is shown in Fig. 3.

Table 6 gives the results of the BERT-based model on SNIPS and ATIS. An examination of the results shows that the *Joint BERT* model improves the SLU performance significantly, which reflects the superiority of BERT. In addition,

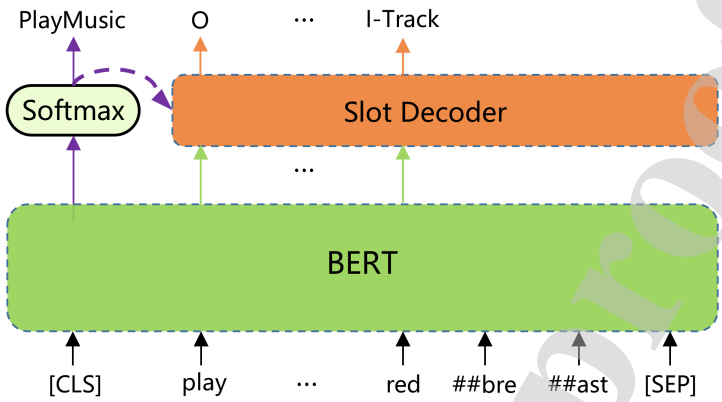


Fig. 3 Illustration of our slot decoder based on BERT. The sample utterance is "*play the song little robin redbreast*".

Table 6 Performance of BERT-based models on two datasets (%).

Model	SNIPS			ATIS		
	Slot(F1)	Intent(Acc)	Overall(Acc)	Slot(F1)	Intent(Acc)	Overall(Acc)
Joint BERT[25]	97.0	98.6	92.8	96.1	97.5	88.2
BERT+Stack-Propa.[10]	97.0	99.0	92.9	96.1	97.5	88.6
BiLSTM+ID+SD	95.2	98.7	88.9	96.0	97.0	87.1
BERT+SD	**97.2**	98.7	**93.2**	**96.3**	**97.6**	88.6

our approach (BERT+SD) achieves the best results on both datasets in terms of overall accuracy, and the performance is improved by 0.4% compared to pure BERT (joint BERT). This finding demonstrates that it is beneficial to explicitly model the relationships between related tasks, particularly when the training data are insufficient. Although the advantage of our approach over the BERT+Stack-Propa is reduced, our approach still outperforms the BERT+StackPropa, which verifies the effectiveness and superiority of our approach.

5.5. Interpretability of Slot Decoder

The interpretability of the proposed slot decoder can be analysed from two effects: 1) structure interpretability and 2) data interpretability. As mentioned above, the proposed slot decoder utilizes the PMMAtt mechanism to explicitly model the interactions between words and intent-slots. Its structure is clear and

reasonable. Previous experiments have also verified that this structure has a positive effect on the expected results. Therefore, the proposed slot decoder is structurally interpretable.

In this section, we mainly discuss the data interpretability. Here, the data refer to the PMMAtt weights, which naturally reflect the interaction between the word encoding features and the intent-slot features, and how the intent-slot features are selected. We will first conduct case studies and then visualize the PMMAtt weights to illustrate the data interpretability of the proposed AISE model.

W	play	the	album	journeyman
	↓	↓	↓	↓
S/T	O	O	B-object_type	B-object_name
S/P	O	O	B-object_type	B-album
I	SearchCreativeWork			

(a)

W	find	on	dress	parade
	↓	↓	↓	↓
S/T	O	B-movie_name	I-movie_name	I-movie_name
S/P	O	B-object_name	I-movie_name	I-movie_name
I	SearchScreeningEvent			

(b)

Fig. 4 Conflicts between slot and intent

In the error analysis, we find that there are two common conflicts in the results predicted by implicit joint models: 1) The conflict between slot and intent, as depicted in Fig. 4(a), where the intent SearchCreativeWork only has three kind of slots, namely, O, object_name, and object_type, but the B-album is predicted; and 2) The conflict between adjacent slots, as shown in Fig. 4(b), where the B-object_name should be followed by O or I-object_name, but the I-movie_name follows it.

The proposed slot decoder can effectively solve the above two conflicts. As shown in Fig. 5, the PMMAtt weights under three heads learned by the slot decoder are transformed into three heatmaps. In each heatmap, the darkness of each color cell is the corresponding PMMAtt weight and the weight sum corresponding to each column of color cells is 1. In the first heatmap, we observe that the start slot of a slot chunk, that is, the slot labelled O or beginning with B, is highly dependent on the intent feature; the middle slots of a slot chunk, whose label start with I, attend to the slot feature at the adjacent position. These

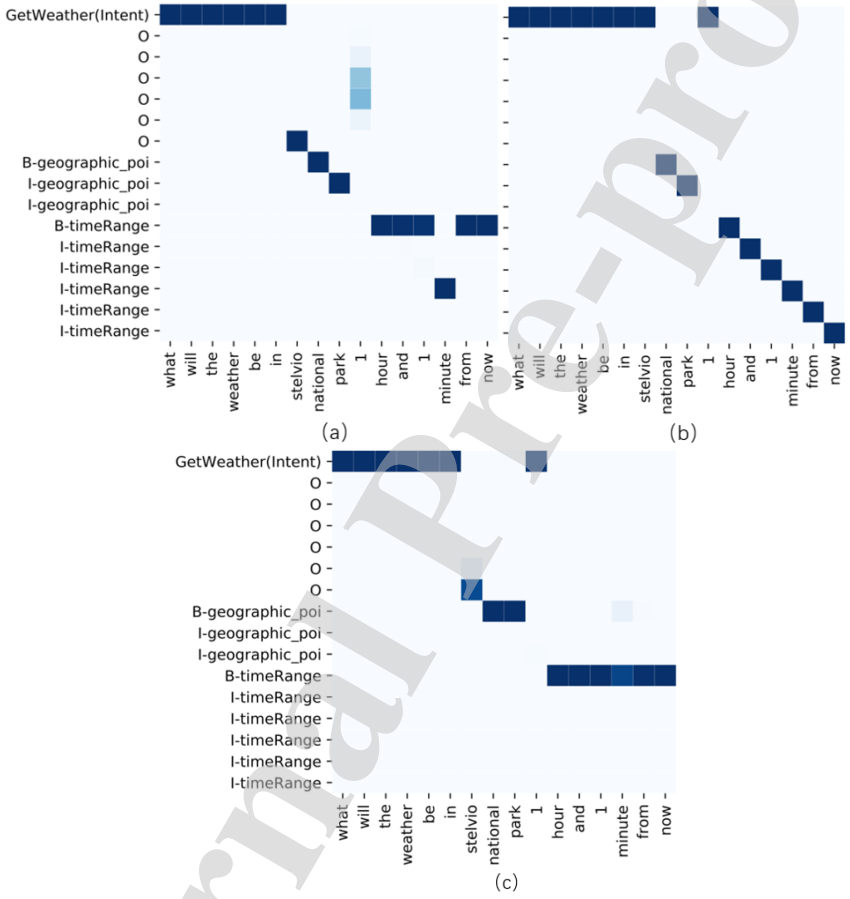


Fig. 5 Heatmaps of learned PMMAtt weights between words (x-axis) and intent-slots (y-axis). An utterance from SNIPS test set is given. The slot of the last word "now" is I-timeRange, which is not shown.

observations are consistent with our intuition, implying that the slot decoder can learn such rules effectively.

Moreover, the model can capture different aspect of features through multiple attention heads. An examination of the second and third heatmaps in Fig. 5 shows that the distribution of weights has changed, where the middle slots of each slot chunk directly focus on the start slot in the slot chunk instead of attending to the features at the adjacent positions. However, the intent feature still plays an important role in the prediction of each slot.

6. Conclusion

In this paper, we propose a novel PMMAtt-based model for joint intent detection and slot filling. In our model, multi-head attention mechanism is introduced to capture utterance-level intent context features, and the PMMAtt mechanism is introduced to model the explicit interactions between word encoding features and intent-slot features. The experimental results show that our model is effective and achieves state-of-the-art results. In addition, we analyse the effect of different model components. Experiments with BERT and different pooling operations further demonstrate the effectiveness of the proposed slot decoder, and the visualization of learned weights illustrates the interpretability of our model. In the future, we plan to incorporate additional knowledge and explore new architectures that can better model the interactions among words, intents, slots and other features for better SLU.

Acknowledgments

This work was supported in part by the Consulting Project of Chinese Academy of Engineering under Grant No. 2020-XY-5, in part by the National Natural Science Foundation of China under Grant Nos. 61472080 and 61672155, and in part by the Collaborative Innovation Center of Novel Software Technology and Industrialization.

References

[1] Y.-Y. Wang, L. Deng, A. Acero, Semantic Frame Based Spoken Language Understanding, Wiley, 2011, pp. 35–80.

[2] G. Tur, L. Deng, Intent Determination and Spoken Utterance Classification, Wiley, 2011, pp. 81–104.

[3] X. Zhang, H. Wang, A joint model of intent determination and slot filling for spoken language understanding., in: IJCAI, Vol. 16, 2016, pp. 2993–2999.

[4] D. Hakkani-Tür, G. Tür, A. Celikyilmaz, Y.-N. Chen, J. Gao, L. Deng, Y.-Y. Wang, Multi-domain joint semantic frame parsing using bi-directional rnn-lstm., in: Interspeech, 2016, pp. 715–719.

[5] C.-W. Goo, G. Gao, Y.-K. Hsu, C.-L. Huo, T.-C. Chen, K.-W. Hsu, Y.-N. Chen, 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, Volume 2 (Short Papers), 2018, pp. 753–757.

[6] C. Li, L. Li, J. Qi, A self-attentive model with gate mechanism for spoken language understanding, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018, pp. 3824–3833.

[7] J. Cheng, L. Dong, M. Lapata, Long short-term memory-networks for machine reading, arXiv preprint arXiv:1601.06733.

[8] S. Zhu, K. Yu, Encoder-decoder with focus-mechanism for sequence labelling based spoken language understanding, in: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2017, pp. 5675–5679.

[9] B. Liu, I. Lane, Attention-based recurrent neural network models for joint intent detection and slot filling, arXiv preprint arXiv:1609.01454.

[10] L. Qin, W. Che, Y. Li, H. Wen, T. Liu, A stack-propagation framework with token-level intent detection for spoken language understanding, arXiv preprint arXiv:1909.02188.

[11] P. Haffner, G. Tur, J. H. Wright, Optimizing svms for complex call classification, in: 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03)., Vol. 1, IEEE, 2003, pp. I–I.

[12] R. E. Schapire, Y. Singer, Boostexter: A boosting-based system for text categorization, Machine learning 39 (2-3) (2000) 135–168.

[13] T. Mikolov, M. Karafiát, L. Burget, J. Černockỳ, S. Khudanpur, Recurrent neural network based language model, in: Eleventh annual conference of the international speech communication association, 2010.

[14] Y. Kim, Convolutional neural networks for sentence classification, arXiv preprint arXiv:1408.5882.

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in neural information processing systems, 2017, pp. 5998–6008.

[16] S. Sabour, N. Frosst, G. E. Hinton, Dynamic routing between capsules, in: I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems 30, Curran Associates, Inc., 2017, pp. 3856–3866.

[17] G. E. Hinton, S. Sabour, N. Frosst, Matrix capsules with em routing, in: International conference on learning representations, 2018.

[18] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, J. Gao, Deep learning based text classification: A comprehensive review, arXiv preprint arXiv:2004.03705.

[19] J.-K. Kim, G. Tur, A. Celikyilmaz, B. Cao, Y.-Y. Wang, Intent detection using semantically enriched word embeddings, in: 2016 IEEE Spoken Language Technology Workshop (SLT), IEEE, 2016, pp. 414–419.

[20] X. Congying, Z. Chenwei, Y. Chenwei, et al., Zero-shot user intent detection via capsule neural networks [c], in: Proc. of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018, pp. 3090–3099.

[21] Z. Tan, M. Wang, J. Xie, Y. Chen, X. Shi, Deep semantic role labeling with self-attention, arXiv preprint arXiv:1712.01586.

[22] T. Mikolov, E. Grave, P. Bojanowski, C. Puhrsch, A. Joulin, Advances in pre-training distributed word representations, arXiv preprint arXiv:1712.09405.

[23] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer, Deep contextualized word representations, arXiv preprint arXiv:1802.05365.

[24] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805.

[25] Q. Chen, Z. Zhuo, W. Wang, Bert for joint intent classification and slot filling, arXiv preprint arXiv:1902.10909.

[26] C. Zhang, Y. Li, N. Du, W. Fan, P. S. Yu, Joint slot filling and intent detection via capsule neural networks, arXiv preprint arXiv:1812.09471.

[27] E. Haihong, P. Niu, Z. Chen, M. Song, 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, 2019, pp. 5467–5471.

[28] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (8) (1997) 1735–1780.

[29] J. Gong, X. Qiu, S. Wang, X. Huang, Information aggregation via dynamic routing for sequence encoding, arXiv preprint arXiv:1806.01501.

[30] Z. Lin, M. Feng, C. N. d. Santos, M. Yu, B. Xiang, B. Zhou, Y. Bengio, A structured self-attentive sentence embedding, arXiv preprint arXiv:1703.03130.

[31] P. Shaw, J. Uszkoreit, A. Vaswani, Self-attention with relative position representations, arXiv preprint arXiv:1803.02155.

[32] Z. Dai, Z. Yang, Y. Yang, W. W. Cohen, J. Carbonell, Q. V. Le, R. Salakhutdinov, Transformer-xl: Attentive language models beyond a fixed-length context, arXiv preprint arXiv:1901.02860.

[33] A. Coucke, A. Saade, A. Ball, T. Bluche, A. Caulier, D. Leroy, C. Doumouro, T. Gisselbrecht, F. Caltagirone, T. Lavril, et al., Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces, arXiv preprint arXiv:1805.10190.

[34] C. T. Hemphill, J. J. Godfrey, G. R. Doddington, The atis spoken language systems pilot corpus, in: Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27, 1990, 1990.

[35] Y. Wang, Y. Shen, H. Jin, A bi-model based rnn semantic frame parsing model for intent detection and slot filling, arXiv preprint arXiv:1812.10235.

[36] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980.

Credit author statement

Paper: KNOSYS-D-20-00951
Title: Attending to Intent and Slots Explicitly for Better Spoken Language Understanding Knowledge-Based Systems
Authors: Peng Yang, Dong Ji, Chengming Ai, Bin Li

Roles:

Peng Yang: Conceptualization, Methodology, Supervision.
Dong Ji: Conceptualization, Methodology, Software, Writing- Original draft preparation.
Chengming Ai: Data curation, Software, Visualization, Investigation.
Bing Li*:* Supervision, Validation, Writing- Reviewing and Editing.

Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: