electronics

Article

A Typed Iteration Approach for Spoken Language Understanding

Yali Pang *, Peilin Yu and Zhichang Zhang

College of Computer Science and Engineering, Northwest Normal University, Lanzhou 730070, China
* Correspondence: pyl@nwnu.edu.cn

Abstract: A spoken language understanding (SLU) system usually involves two subtasks: intent detection (ID) and slot filling (SF). Recently, joint modeling of ID and SF has been empirically demonstrated to lead to improved performance. However, the existing joint models cannot explicitly use the encoded information of the two subtasks to realize mutual interaction, nor can they achieve the bidirectional connection between them. In this paper, we propose a typed abstraction mechanism to enhance the performance of intent detection by utilizing the encoded information of SF tasks. In addition, we design a typed iteration approach, which can achieve the bidirectional connection of the encoded information and mitigate the negative effects of error propagation. The experimental results on two public datasets ATIS and SNIPS present the superiority of our proposed approach over other baseline methods, indicating the effectiveness of the typed iteration approach.

Keywords: spoken language understanding; intent detection; slot filling; typed iteration

Citation: Pang, Y.; Yu, P.; Zhang, Z. A Typed Iteration Approach for Spoken Language Understanding. *Electronics* **2022**, *11*, 2793. https://doi.org/10.3390/electronics11172793

Academic Editor: José L. Abellán

Received: 17 August 2022
Accepted: 3 September 2022
Published: 5 September 2022

1. Introduction

Spoken language understanding is the core module in a task-oriented dialog system. It typically consists of two subtasks: intent detection (ID) and slot filling (SF) [1,2]. For the sentence given by the user, the SLU module identifies the user intent and the utterance slot labels and then uses them to execute user commands or continue a conversation with the user through the dialog management module. For instance, Table 1 shows an example sentence "*Flights from Beijing to Shanghai*" extracted from the ATIS corpus. Each word in the sentence corresponds to a slot label, and the entire sentence has a specific intent. Correctly identifying the intent of the sentence and each word's slot label are essential for downstream tasks.

Table 1. An example sentence.

Sentence	Flights	from	Beijing	to	Shanghai
Slot	O	O	B-fromloc	O	B-toloc
Intent			atis_flight		

Traditionally, the ID and SF tasks are implemented separately by pipeline models. Because the independent pipeline models ignore the characteristics that these two tasks are always co-occurring and highly correlated, the recent trend is to design joint models, which can be seen as a multi-task learning framework for ID and SF. Some studies have proven that joint models are effective in SLU [3–5]. However, the link methods of these joint models are implicit, such as applying a joint loss function or shared embedding. A more effective way is to use the encoded information to inter-act and enhance explicitly [6,7]. In theory, there are two different ways to achieve this explicit improvement. The first is to enhance the SF task by the encoded information of the ID task (I2S), and the second is to enhance the ID task by the encoded information of the slot filling task (S2I). Due to the

poor performance of the S2I method, existing research mostly uses the I2S method, and the reasons for the poor performance of the S2I method have not been explored. In this paper, we believe that the main reason for the poor performance of the S2I method is that part of the slot encoded information can mislead the ID task. If this information can be eliminated, the S2I method is also effective. The main reasons for misleading information are as follows:

1. Due to the variety of slot labels, there are many out-of-vocabulary (OOV) words in some slot categories (contact names, song names, etc.), such as in Case 1, shown in Table 2: the joint model may not know what Adele means. The encoded information of the slot filling task is inaccurate at this time, and in-accurately encoded information can negatively affect the ID task.
2. The specific meanings of the corresponding words in the slot can mislead the ID task (message text, recording content, etc.), as Case 2 shows in Table 2. The meaning of the message text slot in the two example sentences is entirely different, and the encoded information of the slot filling task is also different. However, their slot labels are the same and should obtain the exact representation for the ID task. Understanding the specific meaning of slot words cannot help the ID task but will mislead it.

Table 2. Two examples of misleading information; different colors correspond to different slots. It is still easy to identify the intent only by the Result of Typed Abstraction section, containing only slot categories and not specific slot words.

Case	Sentence	Result of Typed Abstraction	Intent
1	Play Adele's Rolling in the deep. Play Taylor Swift's Sparks Fly.	Play the artist's song	Play music
2	Send a message to Mr. Jack saying tonight Miss White has a date with him. Send a message to Mrs. Tom saying remember to book two Taylor Swift concert tickets.	Send a message to name saying message_text	Send message

Aiming at the problem of misleading information in the S2I method, we propose a typed abstraction mechanism. This mechanism abstracts slot words into type words through the result of the slot filling task and generates an abstract sentence. In this way, the misleading information contained in the slot words can be removed, and only the information that is meaningful to the ID task is retained. Through this mechanism, we can achieve the S2I method and avoid the negative impact of misleading information on the ID task. The Result of the Typed Abstraction section in Table 2 shows the abstract sentences after the typed abstraction mechanism.

Now, we can achieve the S2I method by the typed abstraction mechanism, but this mechanism brings a new problem. Because the mechanism is based on the SF task, it is as vulnerable to error propagation as the pipeline model. To solve this problem, we designed a typed iteration approach. Because the information interaction process is iterative in this approach, the adverse effects caused by error propagation can be mitigated. In addition, the typed iteration approach is not like the previous model that only used a single encoded information interaction method. We used both I2S and S2I methods to achieve interaction explicitly and accomplish a two-way connection of two subtasks.

To summarize, our main contributions are as follows:

3. We propose a typed abstraction mechanism. Through this mechanism, we can use the encoded information of the slot filling task to explicitly enhance the ID task while avoiding the negative impact of misleading information on the ID task.
4. We designed a typed iteration approach. This approach uses the typed abstraction mechanism and feature fusion mechanism to realize the bidirectional connection of the encoded information in the joint model. At the same time, it reduces the

error propagation problem that the typed abstraction mechanism may bring through the iterative process. To the best of our knowledge, this is the first attempt to use the encoded information of two subtasks for simultaneous explicit enhancement and interaction.

5. We performed experiments on two public corpora, ATIS and SNIPS, and our approach achieved state-of-the-art performance on both corpora. The experimental results prove the effectiveness of our model.

2. Related Work

SLU generally includes two subtasks: ID and SF. The traditional implementation method of SLU is to carry out ID subtask first and then SF independently in a pipeline way. Among them, the ID subtask is generally regarded as a classification problem, which is implemented by machine learning approaches such as support vector machines (SVMs) [8] and recurrent neural networks (RNNs) [9]. The SF task is regarded as a sequence annotation problem, which is realized by machine learning methods such as conditional random fields (CRF) [10], convolutional neural net-works (CNNs) [11], and recurrent neural networks (RNNs) [12]. Wang et al. [13] proposed a dual-model learning method for slot filling problem to leverage the large amounts of unlabeled data in a weakly supervised learning manner. The advantage of these methods is that the two subtasks are completed independently, and the goal of the subtasks is clear and easy to implement. The disadvantage is that the interaction and correlation between the two subtasks are not considered.

Therefore, to give full play to the interaction and promotion of the two subtasks, some studies have proposed various joint learning models to simultaneously train and implement ID and SF subtasks. Zhang et al. [4] proposed an RNN-based collaborative model to integrate the correlation between intention and slots. Hakkani et al. [14] proposed a joint RNN model of SF and ID. Liu et al. [5] presented a joint learning model of SF and ID based on the attention mechanism. Goo et al. [15] designed a slot gate mechanism to model the internal relationship between ID and SF. Li et al. [16] proposed an intention-enhanced gate mechanism to fuse the related semantic information between slots and intent. Want et al. [17] proposed a bidirectional model to cross-impact between the user intent and utterance slots. Zhang et al. [18] presented a capsule neural network model to encode hierarchical semantic relations between the slots and the intent of the utterance. E et al. [7] presented an SF-ID neural net-work association model to combine SF and ID subtasks. Hui et al. [19] proposed a continual learning interrelated model to utilize semantic information with different characteristics and balance the performance between ID and SF. Wang et al. [20] presented a transformer-based architecture for ID and SF, which encodes syntactical knowledge in it.

However, the methods mentioned above in general utilize the correlation of the two subtasks mainly based on the joint learning loss of two subtasks and cannot explicitly use the encoded information of the two subtasks to employ mutual interaction. Therefore, we utilized the encoded information of the SF task to augment ID performance. Due to the popularity of pre-trained language models such as BERT [21] and some research having already shown the utility of pre-trained language models on the SLU and other natural language-processing tasks [22,23], we also used BERT as the encoding layer and used the result of the BERT-joint method as the baseline in this paper.

3. Approach

In this section, we introduce our approach in detail. Figure 1 gives an overview of our approach. First, we present the typed abstraction mechanism; this part is the implementation of the S2I method. Second, we introduce our typed iteration approach. Finally, we introduce the joint training method used in our approach.

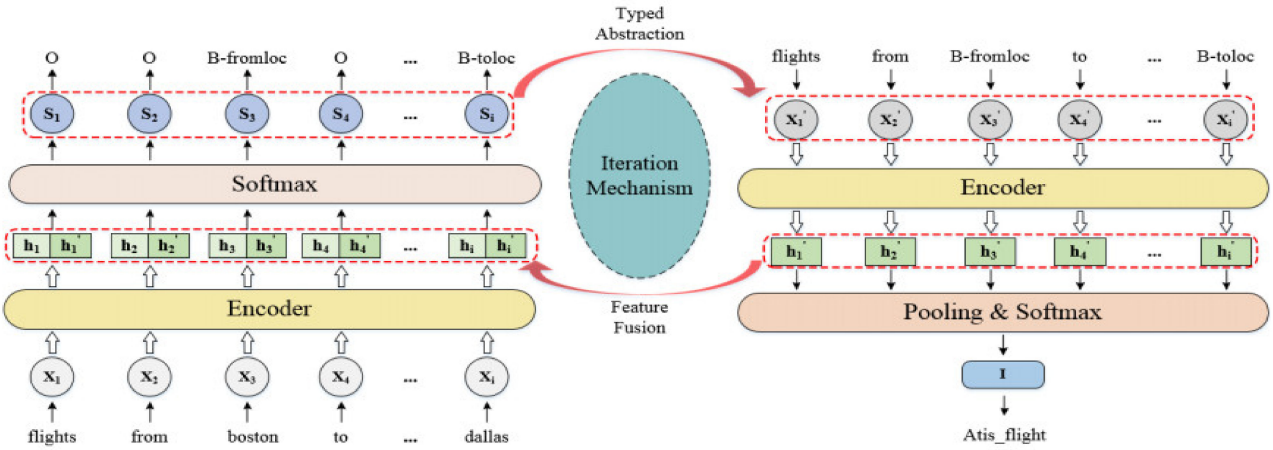

Figure 1. Illustration of our proposed approach. The red arrow indicates that the two subtasks are enhanced and interact through two different mechanisms. We use the same encoder to retain the characteristics of the joint model and use the iteration mechanism to ensure that the process of encoding information interaction is not affected by error propagation.

3.1. Typed Abstraction Mechanism

For a given sentence $X = (w_1, w_2, \ldots, w_n)$, we abstract the sentence based on the results of the SF task $S = (s_1, s_2, \ldots, s_n)$. The specific approach is to use slot filling result S to replace the actual slot words in the sentence with abstract type words and obtain the typed sentence X'. Since the slot words in the typed sentence X' are replaced by type words, they will not be affected by misleading or redundant information when used for intent detection. In order for the two subtasks to interact directly with the encoded information in the approach, we align the sentence during the type abstraction to ensure that X and X' have the same length. The specific process of the type abstraction mechanism is shown in Figure 2.

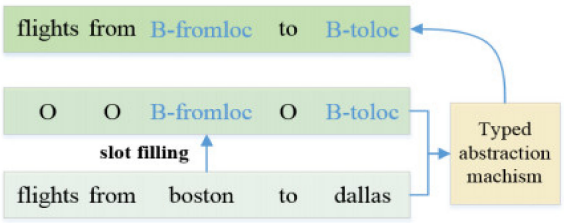

Figure 2. The implementation of a typed abstraction mechanism. It replaces slot words in the input text with abstract type words through the results of the SF task.

3.2. Typed Iteration Approach

The typed iteration approach we propose, as shown in Figure 1, is mainly composed of the following four modules: SF module, typed abstract module, intent-detection module, and feature fusion module. First, in view of the outstanding effect of the pre-trained model BERT [21] on many other natural language-processing tasks, it is used to encode the sentence $X = (w_1, w_2, \ldots, w_n)$. We take the last hidden layer vector $H = (h_1, h_2, \ldots, h_n) \in R^{n \times d}$ (n is the length of the sentence, and d is the dimension of the hidden layer vector) calculated by BERT as an encoding representation of a sentence:

$$H = BERT(w_1, w_2, \ldots, w_n) \tag{1}$$

Then, we perform feature fusion with H and the vector $H' = (h'_1, h'_2, \ldots, h'_n) \in R^{n \times d}$ obtained in the previous iteration (H' in the first iteration is a randomly initialized vector),

to obtain a new hidden layer vector $H^S \in R^{n \times d}$. Then, the vector is used to obtain the SF task result $S = (s_1, s_2, \ldots, s_n)$:

$$H^S = H \oplus H' \tag{2}$$

$$Y^S = \text{softmax}(w^S H^S + b^S) \tag{3}$$

$$S = \text{argmax}(Y^S) \tag{4}$$

After that, the SF result S is used to perform type abstraction mechanism on the input sentence X to obtain the typed sentence $X' = (w'_1, w'_2, \ldots, w'_n)$ and then use it for the intent-detection task. We use the same BERT model to encode X' and obtain the encoded representation $H' = (h'_1, h'_2, \ldots, h'_n) \in R^{n \times d}$ of the typed sentence X'. We processed the vector H' by mean pooling and H^l and then used it to obtain the ID results:

$$X' = \text{TypedAbstraction}(X, S) \tag{5}$$

$$H' = \text{BERT}(w'_1, w'_2, \ldots, w'_n) \tag{6}$$

$$H^l = \frac{1}{n} \sum\nolimits_{x \in n} h'_x \tag{7}$$

$$Y^l = \text{softmax}(w^l H^l + b^l) \tag{8}$$

The ID result $I = \text{argmax}(Y^l)$ and the SF result S are then compared with the results of the previous iteration. If the results are the same, or the preset maximum iteration round is reached, the iteration is terminated, and the result of the current round of ID and SF is used as outputs; otherwise, the above process is repeated. The specific description of the typed iteration algorithm is as shown in Algorithm 1.

Algorithm 1 Typed iteration algorithm

Input: Input sentence X;
Parameter: A random vector H' has the same shape with H; Max iteration number m; t: 0; A vector S_{t-1} initialized to 0; A vector I_{t-1} initialized to 0
Output: S, I
flag←*True*
H = BERT(X)
while *flag* **do**
$S_t = \text{SlotFilling}([H\prime H'])$
$X' = \text{TypedAbstraction}(X, S_t)$
$H' = BERT(X')$
$H^l = \text{MeanPooling}(H')$
$I_t = \text{IntentDection}(H^l)$
if ($S_t == S_{t-1}$ and $I_t == I_{t-1}$ or $t == m$ **then**
$flag \leftarrow False$
else
$S_{t-1} \leftarrow S_t$
$I_{t-1} \leftarrow I_t$
$t \leftarrow t + 1$
end if
end while
return S; I

The ID result I and the SF result S are then compared with the results of the previous iteration. If the results are the same, or the preset maximum iteration round is reached, the iteration is terminated, and the result of the current round of ID and SF is used as outputs; otherwise, the above process is repeated. The specific description of the typed iteration algorithm is shown in Algorithm 1.

3.3. Joint Training

Because the typed abstraction mechanism interrupts the continuity of the gradient, we calculate the loss for each iteration in the typed iteration approach and finally add the losses for joint training. ID and SF loss are calculated as follows:

$$L_S \triangleq -\sum_{x=1}^{t}\sum_{z=1}^{n} \hat{y}_z^S \log(y_z^{S,x}) \tag{9}$$

$$L_I \triangleq -\sum_{x=1}^{t} \hat{y}^I \log(y^{I,x}), \tag{10}$$

where $\hat{y}^I$ and $\hat{y}_z^S$ are the gold intent label and gold slot label, respectively; n is the number of slot labels, and t is the number of iterations. To jointly model the intent classification and SF tasks, the objective is formulated as follows:

$$p(y^S, y^I|X) = p(y^I|X)\prod_{t=1}^{T} p(y_t^S|X) \tag{11}$$

Compared with pipeline models [4], the shared representations learned through joint training by the encoder will combine two subtasks jointly and further ease the error propagation.

4. Experiments

4.1. Datasets

We performed experiments on two public datasets, ATIS [24] and SNIPS [25]. ATIS is a dataset related to the aviation field that is widely used in the task of spoken language understanding. SNIPS is a dataset collected from Snips Voice Assistant, and the distribution of categories is relatively balanced. The specific statistics of the two datasets are shown in Table 3. The format and division of datasets used in this paper are the same [15].

Table 3. Specific statistics of the two datasets.

Dataset	ATIS	SNIPS
Vocabulary size	722	11,241
#Intents	21	7
#Slots	120	72
#Training samples	4478	13,084
#Validation samples	500	700

4.2. Evaluation Metrics

In order to compare the performance of our model with other related work, conveniently, we used three evaluation metrics in the experiments following [6,15]. For the ID task, we used the accuracy rate. For the SF task, the F1-score was applied. In addition, we used the sentence accuracy rate as the overall assessment of the sentence level; that is, the sentence is correct when the intent detection and all slots in a sentence are correctly identified. The calculation method of the evaluation metrics in this paper is the same as [6,15].

4.3. Experimental Settings

We used BERT-Base-Uncased (https://github.com/google-research/bert, accessed on 16 August 2022), released by Google, as the pre-training parameters of our BERT encoding layer. BERT-Base-Uncased is pre-trained based on Bookcorpus (800M words) and English Wikipedia (2500M words), the hidden layer dimension is 768, and the parameter size is 110M. The maximum number of iterations of the typed iteration approach is set to 4, the learning rate is preset to 10^{-5}, and adaptive adjustment is made according to the loss of the validation set.

4.4. Overall Results

Table 4 shows the experimental results of the proposed models on the SNIPS and ATIS datasets. It can be seen that our approach achieves state-of-the-art performance in the three evaluation metrics of both datasets. To prove that the improvement of our approach does not depend on using BERT, we used the same pre-training parameters to reimplement two other models using BERT as the coding layer. From the table, we can see that compared with the best prior stack-propagation + BERT, in the SNIPS dataset, we obtained a 0.3% improvement on intent (Acc), the same performance on slot (F1) score, and 0.4% improvement on sentence (Acc). In the ATIS dataset, we obtained 0.8% improvement on intent (Acc), 0.1% improvement on slot (F1) score and 0.4% improvement on sentence (Acc).

Table 4. Comparison of experimental results of different methods.

Dataset	SNIPS			ATIS		
	Intent (acc)	Slot (F1)	Sentence (acc)	Intent (acc)	Slot (F1)	Sentence (acc)
Joint Seq (Dilek et al., 2016) [14]	96.9	87.3	73.2	92.6	94.3	80.7
Attention BiRNN (Liu et al., 2016) [14]	96.7	87.8	74.1	91.1	94.2	78.9
Slot-Gated Full Atten (Goo et al., 2018) [15]	97.0	88.8	75.5	93.6	94.8	82.2
Slot-Gated Intent Atten (Goo et al., 2018) [15]	96.8	88.3	74.6	94.1	95.2	82.6
Self-Attentive Model (Li et al., 2018) [16]	97.5	90.0	81.0	96.8	95.1	82.2
Bi-Model (Wang et al., 2018) [17]	97.2	93.5	83.8	96.4	95.5	85.7
CAPSULE-NLU (Zhang et al., 2019) [18]	97.3	91.8	80.9	95.0	95.2	83.4
SF-ID Network (Haihong et al., 2019) [7]	97.0	90.5	78.4	96.6	95.6	86.0
Stack-propagation (Qin et al., 2019) [6]	98.0	94.2	86.9	96.9	95.9	86.5
BERT-baseline(Chen et al., 2019) [22]	98.1	96.3	90.4	97.1	85.9	87.9
Stack-propagation + BERT (Qin et al., 2019) [6]	98.6	96.7	91.8	97.3	96.1	88.2
Our model	**98.9**	**96.7**	**92.2**	**98.1**	**96.2**	**88.7**

In addition, from Table 4, we can see that our model performs better on intent-detection tasks due to the role of typed abstraction mechanisms. The performance of the SF task proves that our typed iteration approach avoids the impact of error propagation on the SF task. The result also verifies our assumptions that part of the slot encoded information can mislead the intent-detection task. If this information can be eliminated, the S2I method is also effective.

4.5. Ablation Experiment

In this section, we prove the effectiveness of each part of our method by comparing adding separate modules to the BERT-baseline method, as shown in Table 5.

Table 5. Results of the ablation experiment.

Dataset	SNIPS			ATIS		
	Intent (acc)	Slot (F1)	Sentence (acc)	Intent (acc)	Slot (F1)	Sentence (acc)
BERT-baseline	98.1	96.3	90.4	97.1	95.9	87.9
+Typed Abstraction(TA)	98.4	96.3	90.6	97.3	95.8	88.0
+Feature Fusion(FF)	98.0	96.1	89.9	97.1	95.4	87.5
+TA&FF(M = 1)	98.4	96.5	91.1	97.5	95.9	88.1
M = 3	98.7	96.5	91.5	97.8	95.9	88.4
M = 5	**98.9**	96.6	**92.2**	**98.1**	96.1	88.6
FF by Mean Pooling	98.6	96.4	91.8	97.8	95.9	88.4
FF by Max Pooling	98.6	96.5	92.0	97.7	95.6	86.1
FF by Attention	98.8	**96.7**	**92.2**	**98.1**	96.1	88.7
Our model	**98.9**	**96.7**	**92.2**	**98.1**	**96.2**	**88.7**

The first part of the table shows the effect of adding two interactive methods separately. It can be seen that using only the type abstraction mechanism will slightly increase the intent-detection task, but using the feature fusion mechanism alone will not improve the performance. Using two information interaction methods simultaneously can achieve performance improvement on both subtasks. In the second part, we compared the performance of the approach under different maximum iterations. It can be seen that different maximum iteration numbers will affect the performance of the approach. The selection of the maximum number of iterations will be further described below. In the third part, we compared the effect of different feature fusion mechanisms on performance. It can be seen that the pooling method is worse than the attention mechanism and concatenate method, and the performance gap between the attention mechanism and the concatenate method is not significant. However, the attention mechanism will take more time and memory space, so we chose the concatenate method to achieve feature fusion.

5. Analysis

5.1. Selection of the Maximum Iteration Number

When choosing the maximum iteration number, we do not treat it as a hyperparameter and determine it by tuning the parameters on the validation set. We believe that a correct maximum iteration number M should ensure that as much data as possible can reach convergence within M times. Table 6 shows the details of the training set on the ATIS dataset when selecting different maximum iteration numbers. It can be seen that when the maximum iteration number $M = 3$, 27.51% of the data still fails to converge, indicating that the iteration process is not complete at this time; when $M = 4$, it can ensure that most of the data reach convergence; when $M = 5$, the convergent data growth is not apparent, and the performance is not improved, but the time spent training will increase significantly. Therefore, the maximum iteration number should be performed in multiple experiments on datasets and determined by observing the data convergence on the training set.

Table 6. Details of different maximum iterations.

Max Iteration Number M	$t \leq M$	$t > M$	Time for Training One Epoch
$M = 3$	72.49%	27.51%	2 m 04 s
$M = 4$	91.94%	7.06%	3 m 11 s
$M = 5$	92.33%	7.06%	3 m 58 s

5.2. Effect of Typed Abstraction Mechanism

To prove the effectiveness of the typed abstraction mechanism, we used a set of heatmaps for comparison and illustration, as shown in Figure 3. We used our model and the BERT-baseline model for comparison and used these two models to predict two contrasting sentences that have the same intent but different slot words. We drew the heat map by the last level of the attention distribution matrix in the intent-detection task.

From Figure 3a,b, it can be seen that although the meaning of the two sentences is the same, the attention distribution of the BERT-baseline method is not the same, and it can be seen that the model treats the date as an essential feature, but many sentences with a different intent in the ATIS dataset contain date features. This proves that the implicit joint model will choose some wrong slot types as the classification features, which will affect the model's performance. Figure 3c is a heat map obtained when using our typed iterative approach to predict the comparative sentences. In our approach, two sentences have the same encoding information in the intent-detection task, and it can be seen that dates and location names are no longer essential features for the intent-detection task. Through the comparison of the heat map, we can see that our approach can filter the impact of the specific meaning of slot words on the intent-detection task and help the model choose the accurate features.

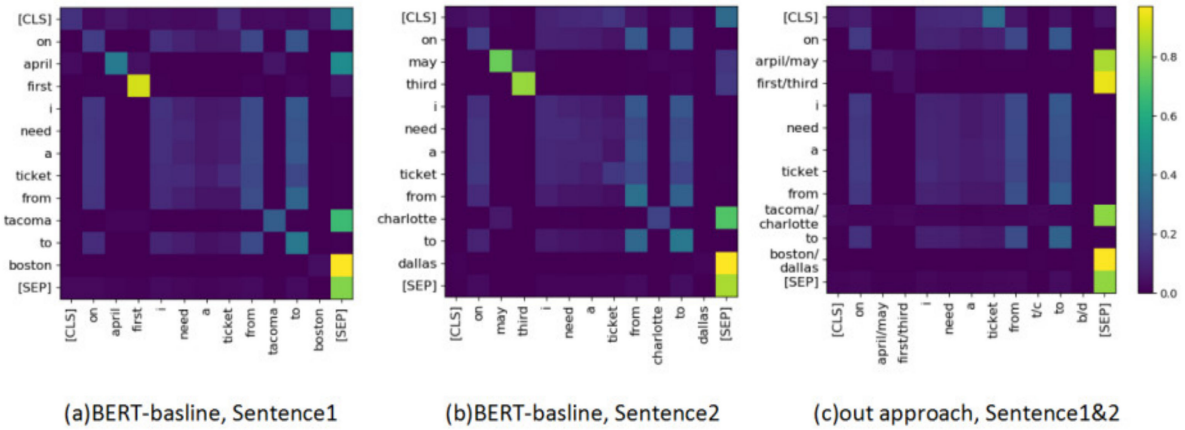

Figure 3. Heat maps comparison of our approach and the BERT-baseline method. It can be seen that our method selects more accurate features in the intent-detection task.

5.3. Case Study

Figure 4 shows the results of the two models on an artificially constructed sentence to compare and illustrate how the typed iteration approach can correct errors through the iterative process. We made two virtual location names, *loc_A* and *loc_B*, to simulate the model's performance when OOV words appear. It can be seen that although the BERT-baseline model can correctly identify the slot labels and intent, the result of time slots produces errors. Our method also makes errors during the first iteration, but it was gradually revised through the iterative process and finally obtained the correct results. This proves the iterative process's necessity and shows that the typed iterative approach is effective when dealing with OOV words.

Sentence		Flight departing from loc_A to loc_B Wednesday after 6 pm
Correct result	Slot	O O O B-fromloc O B-toloc B-depart_date B-depart_time B-depart_time I-depart_time
	Intent	atis_flight
BERT-baseline	Slot	O O O B-fromloc O B-toloc B-arrive_date B-arrive_time B-arrive_time I-arrive_time
	Intent	atis_flight
Our model (t=1)	Slot	O O O B-fromloc O B-toloc B-depart_date B-arrive_time B-arrive_time I-arrive_time
	Intent	atis_flight
Our model (t=2)	Slot	O O O B-fromloc O B-toloc B-depart_date B-depart_time B-depart_time I-depart_time
	Intent	atis_flight

Figure 4. In an example sentence with OOV words, red content indicates an incorrect entity.

6. Conclusions

This paper proposes a typed abstraction mechanism that can explicitly enhance the intent-detection task by the encoded information of the SF task. We also designed a typed iterative approach based on the pre-trained language model BERT. The joint model constructed by this approach can achieve a bidirectional connection of the encoded information of the two subtasks and mitigate the negative effects of error propagation through iteration. We performed experiments on two public datasets, ATIS and SNIPS, and obtained state-of-the-art performance. In addition, through experiments and analysis, we verified the effectiveness of the typed abstraction mechanism and the typed iteration approach.

Author Contributions: Conceptualization, Y.P. and P.Y.; methodology, P.Y.; software, P.Y.; validation, Z.Z. and Y.P.; formal analysis, P.Y.; investigation, Z.Z.; resources, Z.Z.; data curation, Y.P.; writing—original draft preparation, P.Y.; writing—review and editing, Z.Z.; supervision, Z.Z.; project administration, Z.Z.; funding acquisition, Z.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (NO. 62163033); the Natural Science Foundation of Gansu Province, China (NO. 21JR7RA781, NO. 21JR7RA116); Lanzhou Talent Innovation and Entrepreneurship Project, China (NO. 2021-RC-49); and Northwest Normal University Major Research Project Incubation Program, China (NO. NWNU-LKZD2021-06).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Tur, G.; de Mori, R. *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech*; John Wiley & Sons: Hoboken, NJ, USA, 2011.
2. Wu, J.; Harris, I.G.; Zhao, H. Spoken Language Understanding for Task-oriented Dialogue Systems with Augmented Memory Networks. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2021), Online, 6–11 June 2021; pp. 797–806.
3. Chen, Y.N.; Hakanni-Tür, D.; Tur, G.; Celikyilmaz, A.; Guo, J.; Deng, L. Syntax or semantics? Knowledge-guided joint semantic frame parsing. In Proceedings of the 2016 IEEE Spoken Language Technology Workshop (SLT), San Diego, CA, USA, 13–16 December 2016; pp. 348–355.
4. Zhang, X.; Wang, H. A joint model of intent determination and slot filling for spoken language understanding. In Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI-16), New York, NY, USA, 9–15 July 2016; pp. 2993–2999.
5. Bing Liu and Ian Lane. 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 2016), San Francisco, CA, USA, 8–12 September 2016; pp. 685–689.
6. 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 and the 9th International Joint Conference on Natural Language Processing (EMNLP & IJCNLP 2019), Hong Kong, China, 3–7 November 2019; pp. 2078–2087.
7. 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 2019), Florence, Italy, 28 July–2 August 2019; pp. 5467–5471.
8. Haffner, P.; Tur, G.; Wright, J.H. Optimizing svms for complex call classification. In Proceedings of the 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'03), Hong Kong, China, 6–10 April 2003; Volume 1.
9. Lai, S.; Xu, L.; Liu, K.; Zhao, J. Recurrent convolutional neural networks for text classification. In Proceedings of the Twenty-ninth AAAI Conference on Artificial Intelligence (AAAI 2015), Austin, TX, USA, 25–30 January 2015; pp. 2267–2273.
10. Raymond, C.; Riccardi, G. Generative and discriminative algorithms for spoken language understanding. In Proceedings of the Eighth Annual Conference of the International Speech Communication Association (INTERSPEECH 2007), Antwerp, Belgium, 27–31 August 2007; pp. 1605–1608.
11. Xu, P.; Sarikaya, R. Convolutional neural network based triangular CRF for joint intent detection and slot filling. In Proceedings of the 2013 IEEE Workshop on Automatic Speech Recognition and Understanding, Olomouc, Czech Republic, 8–12 December 2013; pp. 78–83.
12. Yao, K.; Peng, B.; Zhang, Y.; Yu, D.; Zweig, G.; Shi, Y. Spoken language understanding using long short-term memory neural networks. In Proceedings of the 2014 IEEE Spoken Language Technology Workshop (SLT), South Lake Tahoe, NV, USA, 7–10 December 2014; pp. 189–194.
13. Wang, J.; Chen, K.; Shou, L.; Wu, S.; Chen, G. Effective Slot Filling via Weakly-Supervised Dual-Model Learning. In Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI-21), Online, 2–9 February 2021; pp. 13952–13960.
14. Hakkani-Tür, D.; Tür, G.; Celikyilmaz, A.; Chen, Y.N.; Gao, J.; Deng, L.; Wang, Y.Y. Multi-domain joint semantic frame parsing using bi-directional rnn-lstm. In Proceedings of the 17th Annual Meeting of the International Speech Communication Association (INTERSPEECH 2016), San Francisco, CA, USA, 8–12 September 2016; pp. 715–719.
15. Goo, C.W.; Gao, G.; Hsu, Y.K.; Huo, C.L.; Chen, T.C.; Hsu, K.W.; Chen, Y.N. 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 2018), Volume 2 (Short Papers), New Orleans, LV, USA, 1–6 June 2018; pp. 753–757.
16. 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 2018), Brussels, Belgium, 31 October–4 November 2018; pp. 3824–3833.

17. Wang, Y.; Shen, Y.; Jin, H. A bi-Model based RNN semantic frame parsing model for intent detection and slot filling. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, New Orleans, LV, USA, 1–6 June 2018; Volume 2, (Short Papers). pp. 309–314.
18. Zhang, C.; Li, Y.; Du, N.; Fan, W.; Yu, P. Joint Slot Filling and Intent Detection via Capsule Neural Networks. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 5259–5267.
19. Hui, Y.; Wang, J.; Cheng, N.; Yu, F.; Wu, T.; Xiao, J. Joint Intent Detection and Slot Filling Based on Continual Learning Model. In Proceedings of the 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 6–11 June 2021; pp. 7643–7647.
20. Wang, J.; Wei, K.; Radfar, M.; Zhang, W.; Chung, C. Encoding Syntactic Knowledge in Transformer Encoder for Intent Detection and Slot Filling. In Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI-21), Online, 2–9 February 2021; pp. 13943–13951.
21. Devlin, J.; Chang, M.; Lee, K.; Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
22. Chen, Q.; Zhuo, Z.; Wang, W. BERT for joint intent classification and slot filling. *arXiv* **2019**, arXiv:1902.10909.
23. Li, C.; Qiu, Z. Targeted BERT pre-training and fine-tuning approach for entity relation extraction. In Proceedings of the International Conference of Pioneering Computer Scientists, Engineers and Educators (ICPCSEE 2021), Communications in Computer and Information Science, Taiyuan, China, 17–20 September 2021; Volume 1452, pp. 116–125.
24. Hemphill, C.T.; Godfrey, J.J.; Doddington, G.R. The ATIS spoken language systems pilot corpus. In Proceedings of the Speech and Natural Language, Hidden Valley, PA, USA, 24–27 June 1990; pp. 96–101.
25. Coucke, A.; Saade, A.; Ball, A.; Bluche, T.; Caulier, A.; Leroy, D.; Doumouro, C.; Gisselbrecht, T.; Caltagirone, F.; Lavril, T.; et al. Snips voice platform: An embedded spoken language understanding system for private-by-design voice interfaces. *arXiv* **2018**, arXiv:1805.10190.