Article

Multitask Learning with Knowledge Base for Joint Intent Detection and Slot Filling

Ting He ¹, Xiaohong Xu ^{1,*}, Yating Wu ², Huazhen Wang ¹ and Jian Chen ³

¹ College of Computer Science and Technology, Huaqiao University, Xiamen 361021, China; heting@hqu.edu.cn (T.H.); wanghuazhen@hqu.edu.cn (H.W.)

² China Construction Bank Xiamen Branch, Xiamen 361001, China; 17013083022@stu.hqu.edu.cn

³ Zoesoft Co., Ltd., Xiamen 361008, China; chenjian@zoesoft.com.cn

* Correspondence: 18014083023@stu.hqu.edu.cn

Abstract: Intent detection and slot filling are important modules in task-oriented dialog systems. In order to make full use of the relationship between different modules and resource sharing, solving the problem of a lack of semantics, this paper proposes a multitasking learning intent-detection system, based on the knowledge-base and slot-filling joint model. The approach has been used to share information and rich external utility between intent and slot modules in a three-part process. First, this model obtains shared parameters and features between the two modules based on long short-term memory and convolutional neural networks. Second, a knowledge base is introduced into the model to improve its performance. Finally, a weighted-loss function is built to optimize the joint model. Experimental results demonstrate that our model achieves better performance compared with state-of-the-art algorithms on a benchmark Airline Travel Information System (ATIS) dataset and the Snips dataset. Our joint model achieves state-of-the-art results on the benchmark ATIS dataset with a 1.33% intent-detection accuracy improvement, a 0.94% slot filling F value improvement, and with 0.19% and 0.31% improvements respectively on the Snips dataset.

Keywords: multitask learning; knowledge base; intent detection; slot filling; task-oriented dialog system

Citation: He, T.; Xu, X.; Wu, Y.; Wang, H.; Chen, J. Multitask Learning with Knowledge Base for Joint Intent Detection and Slot Filling. *Appl. Sci.* **2021**, *11*, 4887. <https://doi.org/10.3390/app11114887>

Academic Editor: Kuen-Suan Chen

Received: 11 April 2021

Accepted: 13 May 2021

Published: 26 May 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the development of the task-oriented dialog system, nature language understanding (NLU), as a critical component of the task-oriented dialog system, has attracted great research attention. We can capture context information to identify the user's intent by using intelligent interactive devices that talk to humans in different scenarios, and extracting the semantic constituents from the text that the user inputs into the semantic slots that were previously defined [1]. These two modules, namely intent detection and slot filling, can convert the text into its semantic representation, which provides the task information for supporting the dialog system and helps users achieve their demands. Intent detection is the task of classifying natural language utterances into semantic intent classes that have been previously defined. Take Siri as an example. A sentence spoken by a user, such as, "Tell me about the weather" should be classified as a weather-query subtype. Several classifiers, such as the support vector machine (SVM) and convolutional neural network (CNN) have been applied to detect a user's intent. Generally, the intent-detection method constructs a classifier based on a lexicon and semantic feature training dataset [2,3]. However, since the text that a user inputs into the task-oriented dialog system is generally short and a large-scale corpus cannot be found that can be used to train a high-quality model, the intent-detection task cannot be adequately fulfilled. Slot filling is used to allocate appropriate semantic tags to each input word by extracting the semantic concept. For instance, for the sentence "I want the cheapest airfare from Tacoma to Orlando", the slot-filling task should tag Tacoma as the departure city and Orlando as the arrival city. It can be treated as a sequence-labeling task that maps an input word sequence to the

corresponding slot-label sequence. Popular approaches to solve the sequence-labeling problem include generative models [4], such as composite hidden Markov models (HMM), and discriminant or conditional models, such as conditional random fields (CRF) [5] and the support vector machine (SVM) [6,7]. Despite many years of research, the slot-filling task is still a challenging problem, and a few neural-network and deep-learning approaches have been used to solve this problem in recent studies.

Intent detection and slot filling have always been adopted as the pipeline modules in traditional task-oriented dialog system research. However, the pipeline methods propagate errors easily. Differently from the pipeline methods, the joint models have the advantages of utilizing the dependency between intents and slots. These works are mainly divided into two categories. One is the use of semantic analysis [8,9]. Another type of joint model is deep learning, which integrates both feature design and classification into the learning procedure [10,11]. But little research has made full use of additional information, such as the incidence relations and the shared resources between intent-detection and slot-filling modules. Moreover, the above studies have a common problem, i.e., that the standardized and high-quality public datasets are very insufficient, because collecting and labelling datasets require a lot of time and effort. Therefore, we attempt to add an external knowledge base to the existing datasets to solve the problem of data scarcity [12]. A knowledge base is a knowledge cluster engineered so that it is structured, easy to operate and use, and contains comprehensive and organized knowledge. In addition, it is a collection of interrelated knowledge pieces that are stored, organized, managed, and used in computer memory by a certain knowledge representation mode for solving problems in some fields. The latest researches show that the effects of NLP can be improved by introducing a knowledge base [13,14]. In recent research, Weld summarized the joint model, which has become the mainstream method, and some scholars have adopted the external knowledge base (knowledge graph) in applied research [15]. Based on this, we introduce a solution for the above problems by improving the loss function to optimize the joint model and verify the validity of an external knowledge base in the joint model with intent detection and slot filling.

In order to solve the above problems, we propose a joint model of intent detection and slot filling based on multitask learning (MTL) with a knowledge base, which makes full use of the external knowledge and the high-quality relationship information between intents and slots. This is particularly appropriate as the latest method to pre-training language representation, bidirectional encoder representations from transformers (BERT) is used to convert text inputted into a vector in this model. Compared with alternative methods, BERT can be applied to extract the semantic features of the text inputted. Furthermore, the shared parameters and features between the two modules are extracted. This part is introduced to make use of mutual promotion and shared resources between the two modules of intent detection and slot filling. Then, the text that the user inputs is too short to fully express information in practice. In order to increase information other than the text, the external knowledge, such as the WordNet knowledge base, is employed as well. To the best of our knowledge, this is the first time that a knowledge base has been introduced to build an intent-detection and slot-filling model. In addition, a weighted-loss function based on a weighted self-learning algorithm is built to optimize the joint model. Considering the strong correlation between the intents and the slots, our model achieves better performance by using external knowledge and mutually promoting the two modules.

Our main contributions are threefold:

1. Sharing information between the intention-detection module and slot-filling module by the multitask learning method, which avoids the problem of error propagation easily caused by the traditional pipeline method.
2. We introduce the external knowledge base, which can effectively improve the defects of the model, such as lack of semantics and poor generalization performance under the training of specific datasets.

3. We establish the weighted-loss function based on the weighted self-learning algorithm, which plays a strong role in promoting the optimization of the joint model and the improvement of accuracy.

2. Related Work

Both the intent-detection and slot-filling modules are usually used to convert the text that a user inputs into the task-specific user's intent and task-specific semantic representation [16].

Previous work mainly viewed intent detection as an utterance classification task and explored various classification methods. A common approach is to build a multiclass classifier that is trained using the lexical and semantic features of the utterances. In addition, with the development of deep learning, deep belief nets (DBNs) have been used for natural language call routing's intent-detection task for the first time, and they have produced better results compared with the traditional model [17]. Although standard classifiers have produced good results for the intent-detection module in several domains, it is not enough to rely on intention labels solely. Some studies look for auxiliary tasks to promote the intent-detection model's implementation. Celikyilmaz presented a probabilistic topic model for identifying hidden semantic intent classes and intent-bearing constituents from spoken language utterances [18]. Ji proposed a variational Bayesian approach for modeling the latent intents of user queries and clicked URLs when available [19]. They used this model to enhance the supervised intent classifications of user queries from conversational interactions. All the above tasks use auxiliary tasks to help the implemented intent detection, but it is easy to incur unnecessary overheads. Troussas constructed an intention-detection model that can identify students' learning styles by using the integrated classification method. It combines three classifiers: SVM, naive Bayes (NB), and K-nearest neighbor (KNN), which are based on majority voting rules to effectively utilize personal characteristics (i.e., age and gender) and cognitive characteristics as the minimum amount of input for model training to determine learning style [20]. Giannakas proposed a binary classification framework based on deep neural networks (DNN) and extracted the most important features that had the positive or negative impact on the final prediction [21].

Slot filling can be treated as a sequence-labeling task. Traditional approaches including the HMM and CRF can solve the problem to some extent, but the F values are poor. Recently, neural network models, such as recurrent neural networks (RNNs), LSTM, and CNNs have been applied to the slot-filling module [22,23]. Xu tried to use a bidirectional-LSTM-CRF (BiLSTM-CRF) in the sequence labeling for the slot-filling task and showed that the BiLSTM-CRF model provides a significant improvement compared with other models [24].

Joint models for intent detection and slot filling have also been explored. These works are mainly divided into two categories. One is the use of semantic analysis. Tur presented a dependency parsing-based sentence simplification approach that extracts a set of keywords from natural language sentences and uses those in addition to entire utterances for completing NLU tasks [8]. Mairesse showed that a semantic tree can be parsed by recursively using discriminative semantic classification models whose outputs were used to recursively construct a semantic tree, resulting in both slot and intent labels [9]. Although great progress has been made, these methods require good feature engineering and even additional semantic resources. Another type of joint model for intent detection and slot filling is deep learning, which integrates both feature design and classification into the learning procedure. Jeong proposed a triangular CRF (TriCRF) that coupled an additional random variable for intent detection with a standard CRF. This structure both encodes the dependencies of intents and slots and preserves the uncertainty between them [10]. Yao improved the RNN by using the transition features and the sequence-level optimization criterion of the CRF to explicitly model the dependencies of output labels [11]. Xu put forward a CNN-TriCRF model for improving the TriCRF model by using a CNN to automatically extract features to simultaneously handle intent detection and slot filling [25]. However, these works lacked the ability to acquire the long-term memory, which is crucial for sequence labeling. Hakkani-Tür proposed an RNN-LSTM framework that completely

estimated the semantic frame by leveraging public data from multiple domains [26]. Liu and Lane consolidated information about the hidden states from an RNN model for slot filling and then generated intent information by using an attention model [27]. This work applied a joint loss function to link the two tasks implicitly. Hua connected and fed the output vector of BiLSTM and the CNN to the CRF layer to accomplish this task [28]. However, this research neglects the shared features of the intent detection and slot-filling modules. Chih-Wen proposed a slot-gated model that introduces an additional gate that leverages the intent context vector for improving slot-filling performance [29]. This model considers the effect of intents on slots, but ignores the influence of the mutual promotion between these two modules. In recent studies, the application of incidence relations has been gradual. Li simultaneously optimized both the tasks via joint learning in an end-to-end manner [30]. Haihong proposed a bi-directional interrelated model for joint intent detection and slot filling [31]. Chen proposed a multihead self-attention joint model with a conditional random field (CRF) layer and a prior mask [32]. The incidence relations between these modules have already been introduced to some extent. In addition, the pre-trained BERT model provides a powerful context-dependent sentence representation, which can serve to obtain semantic information easily. Semantic information is useful to the joint model. There are several studies that have focused on the BERT application. Namely, Chen implements the BERT pre-trained model to address the poor generalization capability of NLU [33,34]. Castellucci adapted the original BERT fine-tuning method to define a new joint-learning framework [35].

Although the joint model can solve the problems in the current research to some extent, previous studies have not made full use of the shared resources between these modules. Furthermore, the knowledge base is of significance for various NLP applications. It can process information of types other than text and may serve to complete a task in the case of limited data. For example, Xu proposed a knowledge-based topic model to extract more meaningful phrases and coherent topics [13]. Alrehamy introduced SemCluster, a clustering-based unsupervised key phrase extraction method that is used to mitigate the coverage limitation problem by considering knowledge of a wider background [14]. However, knowledge base has not been applied to intent-detection and slot-filling modules in recent studies. We consider that the current modeling methods do not make full use of the information, which can be extracted from a knowledge base. Since intents and slots are usually highly dependent on each other, our work focuses on how to establish the relationship between intents and slots based on MTL. At the same time, our model introduces a knowledge base to facilitate the performance improvement of intent detection and slot filling.

3. Multitask Learning with Knowledge Base for Joint Slot-Filling and Intent-Detection

Intent detection and slot filling are important modules in a task-oriented dialog system. These two modules can convert the text that a user inputs into a semantic representation, which provides task information to support the dialog system. However, completely utilizing mutual promotion and shared resources between the two modules are difficult by only relying on joint models that are presently used. And the value of a knowledge base to these modules has not yet been explored. To solve these problems and improve the accuracy of the intent-detection module and the F1 score of the slot-filling module, this paper proposes a joint model based on MTL with a knowledge base. Benefiting from the MTL framework and external knowledge, our model obtains the shared parameters and features between two modules and implements joint optimization by using a weighted-loss function. Moreover, a knowledge base is needed to improve model performance. In Figure 1, we show the flow chart of this model.

Figure 1. The flow chart of our model.

The input of the model is the text x of an utterance, which is a sequence of words— $x_1, x_2, x_3, \dots, x_T$ —and T is the length of the utterance. The model consists of two kinds of output, i.e., the intent label y^I and the slot label sequence y_i^S . Our method includes the following steps. First, we propose a general LSTM–CNN shared presentation layer in which a text sequence sequentially passes through the LSTM and CNN in order to obtain the text shared-representation features. Then, we establish the Bi-LSTM model with the attention mechanism for the intent-detection and slot-filling modules respectively, according to the differences between the intent-label information and the slot label information. Taking both these modules into consideration, WordNet as a knowledge base is introduced in the Bi-LSTM model. This is conducted with the objective of extracting characteristics of each task based on the shared representation features, and modifying the hidden vector of the original Bi-LSTM model via the knowledge base. Afterwards, a weighted-loss function based on the weighted self-learning method is used for joint optimization. Finally, the model is optimized by adaptive moment estimation (Adam).

Next, each layer of the model will be explained in more detail.

3.1. Presentation of the Shared Representation Features

To obtain the temporal order information and feature information of the text that a user inputs, we proposed a general LSTM–CNN shared representation layer in which the text sequence x sequentially passes through the LSTM and CNN. LSTM is a special kind of RNN that has a strong ability to handle the long-term dependency problem. In general, the LSTM unit consists of three thresholds that control the proportion of information that needs to be forgotten during the information-transfer process. The CNN extracts and selects the features to form the representation vectors for each sequence by scanning the input vector. Feature extraction is performed by using a convolution operation. Then, the most significant local feature is extracted and the global feature vector is formed by the ReLU activation function and the maximum-over-time pooling operation.

In this layer, we convert the text that a user inputs into a vector by BERT. BERT is an attention-based architecture to pre-train language representations, and it has subsequently used that model for NLP tasks. BERT outperforms one-hot and word2vec methods because it is the first unsupervised, deeply bidirectional system for NLP pre-training. Then, the vector is fed through the LSTM network to obtain the text timing information, and then it is fed through the CNN network to obtain the sequence information. Thus, the sharing parameter representations $h^{(shared)}$, which will serve as the common features for intent detection and slot filling, are obtained. Finally, the shared parameter representations $h^{(shared)}$ are obtained.

3.2. Development of the Intent-Detection and Slot-Filling Model

The two Bi-LSTM models extract their respective characteristics for each task based on the shared parameter representations $h^{(shared)}$. As shown in Figure 1, the output of the shared presentation layer flows in two parts. One part is the Bi-LSTM model with the attention mechanism that completes the intent detection task, and the other part achieves the slot filling task. Two training datasets $Data_{intent}$ and $Data_{slot}$ are constructed with the intent tag and the slot tag, respectively. We establish the Bi-LSTM models with the attention mechanism for the intent-detection and slot-filling tasks, respectively. The structure of this part is shown in Figure 2. Accordingly, the intent detection that predicts output y^I and the slot filling that predicts output y_i^S are obtained.

Figure 2. The structure of the intent-detection and slot-filling model. The O denotes the “none” type and B-X denotes that the segment in which this token resides is of type X and that this token is at the beginning of the segment.

Taking the text sequence x and $h^{(shared)}$ as the input, the corresponding hidden layer output $h = (h_0, \dots, h_T)$ is obtained, and the attention mechanism is introduced in the hidden layer to calculate the probability distribution of attention att . $att_{i,j}$ is calculated as follows:

$$att_{i,j} = \frac{\exp(e_{i,j})}{\sum_{k=1}^T \exp(e_{i,k})}, \quad (1)$$

$$e_{i,j} = f(h_{i-1}, h_i). \quad (2)$$

After obtaining the probability distribution of attention at any time, the feature vector c that contains the text information is calculated as follows:

$$c^I = \sum_{j=0}^T att_{i,j} h_j. \quad (3)$$

The last part is the output layer. The softmax function is applied to the representations using a linear transformation to obtain the distribution y^I over the intent labels:

$$y^I = \text{softmax}(\mathbf{W}_s h_T + \mathbf{b}_i), \quad (4)$$

where $\mathbf{W}_s$ and $\mathbf{b}_i$ is the weight matrix, and $\mathbf{b}$ represents the offset vector.

The slot-filling maps the text sequence x to its corresponding slot sequence label y_i^S . The Bi-LSTM generates the bidirectional hidden state h_i at time t , which is defined as the concatenation of the forward hidden state and backward hidden state. For each hidden state h_i , the slot context vector c_i^S is calculated as the weighted sums of the Bi-LSTM's hidden states $h_1, \dots, h_T$ using the learned attention weights α . The slot context vector c_i^S can also be calculated in the same way as c^I , and the slot label of the i -th word is expressed as follows:

$$y_i^S = \text{softmax}(\mathbf{W}_s h_i + \mathbf{b}_i). \quad (5)$$

During the training process, the results of the intent-detection and slot-filling model are used for modeling the slot-intent relationships:

$$r = \sum \eta \log(\mathbf{W}_1 c^I + \mathbf{W}_2 c_i^S)^2. \quad (6)$$

This parameter updates y^I and y_i^S to influence the output of the intent-detection and slot-filling modules, where Equations (4) and (5) are reformed as (7) and (8) respectively:

$$y^I = \text{softmax}(\mathbf{W}_s h_T r + \mathbf{b}_i), \quad (7)$$

$$y_i^S = \text{softmax}(\mathbf{W}_s h_i r + \mathbf{b}_i). \quad (8)$$

3.3. Incorporate External Knowledge

In this part, our aim is to take advantage of the knowledge base to extend the Bi-LSTM model. To effectively integrate the knowledge base with information from the text inputted by a user, our model updates the hidden vector to enhance the learning of Bi-LSTM. It is capable of leveraging a knowledge base when it processes each word in the text, and the hidden vector is updated by semantic relevance to the knowledge. At each time step, the model retrieves concepts that are related to the current word as candidate concepts from WordNet. The candidate concepts are transformed to an embedded word using BERT and fed into the model along with the text input by the user.

Specifically, the knowledge at time-step t comprises candidate knowledge, $V(x_t)$, for the text that a user input, x_t . Each candidate knowledge item $i \in V(x_t)$ is associated with a knowledge vector v_i . The attention weight $\alpha_{t,i}$ for vector v_i via a bilinear operator can be described as follows:

$$\alpha_{t,i} = f(\mathbf{W}_v v_i h_t), \quad (9)$$

where $\mathbf{W}_v$ is a parameter matrix to be learned and h_t is the current hidden vector.

Let m_t be considered as a knowledge state vector that encodes external knowledge information with respect to the input at time t . The mixture model is defined below:

$$m_t = \sum_{i \in V(x_t)} \alpha_{t,i} v_i. \quad (10)$$

We combine it with the hidden vector h_t of the Bi-LSTMs to obtain a knowledge state vector h'_t to modify the original hidden vector h_t :

$$h'_t = h_t + m_t. \quad (11)$$

If $V(x_t)$ is null, we set $m_t = 0$. h'_t can be used for predictions in the same manner as the original hidden vector h_t .

3.4. Optimization of Intent Detection and Slot Filling Model

The loss function is used to characterize the error between the output value of the neural network and the true value. The influence of the loss function on the adjustable parameters of the neural network cannot be ignored. If the loss function is not used properly, the parameters of the neural network will not be satisfactory, even if the training is performed many times.

In our joint model, the cross entropy can be used as a loss function in the back propagation of neural networks and plays a role in the network parameters' update.

Particularly, the loss function $Loss_{intent}$ is defined as the cross-entropy of the predicted output y^I and the true intent y^{intent} .

$$Loss_{intent} = - \sum_{i=1}^T y^{intent} \log(y^I). \quad (12)$$

To ensure that the result of each iteration is close to the real label, the loss function of slot filling $Loss_{slot}$ is calculated as the average cross-entropy of the slot-filling model's predicted output y_i^S and the real slot sequence.

$$Loss_{slot} = \frac{1}{T} \sum_{i=1}^T L(y_i^S, y_i^{slot}), L(y_i^S, y_i^{slot}) = - \sum_{i=1}^M y_i^{slot} \log(y_i^S). \quad (13)$$

In Equation (13), y_i^S is the sequence label of the i -th iteration, y_i^{slot} is the true label, T is the total number of iterations, M is the length of sentence, and L is the cross-entropy.

MTL needs to consider the proportion of the loss function of each task. Different proportions determine the importance of the shared information that is provided by different tasks. In the joint training process, the final loss function is the summation of the losses that were obtained by completing different tasks. In this paper, the weighted-loss function is constructed based on taking the proportions of the intent-detection model and the slot-filling model as the weight parameters. The loss function in our joint model is shown in Equation (14).

$$Loss = \alpha Loss_{intent} + \beta Loss_{slot}. \alpha + \beta = 1 \quad (14)$$

In Equation (14), the function $Loss$ is the total loss of the joint model, and α and β are the weight coefficients of the preset intent-detection task and the slot-filling task, respectively.

The gradient descent algorithm is conducive to quickly finding the optimal solution. The setup of α is established by using the weighted self-learning method based on the gradient descent algorithm, with the calculation steps as follows:

$$\begin{aligned} \frac{\partial Loss}{\partial \alpha} &= \sum_x \left(\frac{t}{f(z)} - \frac{1-t}{1-f(z)} \right) \frac{\partial f}{\partial \alpha} \\ &= \sum_x \left(\frac{t}{f(z)} - \frac{1-t}{1-f(z)} \right) f'(z) x \\ &= \sum_x \frac{x f'(z)}{f(z)(1-f(z))} (f(z) - t) \\ &= \sum_x x (f(z) - t) \end{aligned} \quad (15)$$

In Equation (15), $f(z)$ represents the output value of the model, t is the true value of the sample, and $(f(z) - t)$ is the error between the output value and the true value t of the sample. Therefore, when the value of $(f(z) - t)$ is larger, the error is larger, the gradient value is larger, the weight parameter α is adjusted faster, and the training speed is also faster.

With the gradient calculation being performed on the weight parameter, the value α is iteratively updated using Equation (16).

$$\alpha \leftarrow \alpha d \frac{\partial Loss}{\partial \alpha}. \quad (16)$$

In Equation (16), d is expressed as the learning rate of the gradient step.

When the monotonicity of loss cannot be maintained, the iteration is stopped and the value α is obtained.

Finally, we use Adam, an algorithm for the first-order gradient-based optimization of stochastic objective functions, as an optimization method to optimize our model.

4. Experimental Analysis

4.1. Dataset and Knowledge Base

4.1.1. Dataset

To evaluate the model that is proposed in this paper, experiments were conducted on the Airline Travel Information System (ATIS) dataset. The ATIS dataset [36] is the most-used dataset for NLU research. The dataset consists of sentences that people used when they made flight reservations. The training set contains 4978 utterances, and the test set contains 893 utterances. There were 18 different intent types and 127 distinct slot labels in total.

In our experiment, the ATIS dataset was partitioned into a training set, a validation set, and a test set according to a ratio of 7:1:2. The training set was used to train the model, the validation set was used to adjust the hyperparameters, and the test set tested the generalization performance of the model. An example sentence, “What are the flights from Tacoma to San Jose on Wednesday the nineteenth?” is demonstrated in Table 1. This sentence follows the popular IOB (in–out–begin) format for representing the slot tags. The domain of the sentence is airline travel and the intent is to find a flight. The word “Tacoma” is labeled as the departure city and “San Jose” is labeled as the arrival city. In addition, the word “nineteenth” is labeled as the departure date.

Table 1. An example utterance from the ATIS dataset.

Method	Entity	Relation
atis_flight	what	O
	are	O
	the	O
	flights	O
	from	O
	Tacoma	B-fromloc.city_name
	to	O
	San	B-toloc.city_name
	Jose	I-toloc.city_name
	on	O
	Wednesday	B-depart_date.day_name
	the	O
	nineteenth	B-depart_date.day_number

To verify the generality of the proposed model, we used another NLU dataset collected by Snips for model evaluation. This dataset is collected from the Snips personal voice assistant. The training set contains 13,084 utterances and the test set contains 700 utterances. Compared with the single-domain ATIS dataset, Snips is more complicated mainly because

of the intent diver. The number of slot labels and intent type are 72 and 7, respectively. These seven intents include Search Creative Work, Get Weather, Book Restaurant, Play Music, Add to Playlist, Rate Book, and Search Screening Event.

4.1.2. Knowledge Base

We used WordNet as our external knowledge base. WordNet is a semantic knowledge base which describes objects including compound, phrasal verb, collocation (collocation), idiomatic phrase, and word, of which word is the most basic unit. Unlike traditional dictionaries and thesaurus, WordNet has the following three features:

1. WordNet is organized by synonym set (Synset) as the basic building unit, in which users can find an appropriate word to express a known concept.
2. WordNet associates synonym sets with certain relationship types. There are synonymy, antonymy, hypernymy/hyponymy, meronymy, and entailment, etc. WordNet tries to make the relationships between words simple and easy to use.
3. In WordNet, most Synesets have explanatory comments, but a Synset is not equal to a single entry in a dictionary, because a Synset contains only one comment, while an entry in a traditional dictionary is polysemous and can have multiple interpretations.

4.2. Baselines

According to the topic of intent detection and slot filling in a task-oriented dialog system, we tested many studies on ATIS that focus on the intent-detection or slot-filling tasks to serve as baselines.

- **RNN-LSTM:** Hakkani-Tür presented an approach that jointly modeled slot filling, intent detection, and domain classification in a single bidirectional RNN with LSTM cells [26].
- **Attention-based RNN:** Liu and Lane studied using an RNN for the NLU task, with particular attention on modeling the output sequence dependencies [27]. The authors proposed to model the slot-label dependencies using a sampling approach by feeding the sampled output labels back to the sequence state.
- **Slot-gated:** Chih-Wen proposed a slot-gated model that introduces an additional gate that leverages the intent context vector to improve the slot-filling performance [28].
- **CAPSULE-NLU:** C. Zhang proposed a capsule-based neural network model that accomplished slot filling and intent detection via a dynamic routing-by-agreement scheme [37].
- **Joint BERT:** Q. Chen proposed a joint intent-classification and slot-filling model based on BERT [33].

4.3. Experimental Setup

LSTM and CNN are meta modules in our joint model. According to the setup of the CNN, the number of filters in the convolution layer was set to 64 and the size of the kernel was set to 5. We used the ReLU function as the activation function. The convolutional initialization is orthogonal. With respect to the setup of LSTM and Bi-LSTM, the number of units was set to 128 and the number of hidden vectors was set to 64. During the regularized model training, the dropout rate value was set to 0.5 for acyclic connections, and the maximum number of iterations was set to 100 and 120 on the ATIS and Snips datasets respectively. The external knowledge base used WordNet. For the weighted-loss function, we used the weighted self-learning method based on the gradient descent algorithm to identify the weights, and the initial value was defined as 0.1. We used the Adam optimization method to adjust the parameters. Additionally, we used mini-batch training and set the mini-batch size to 16.

4.4. Results

The main task of the MTL with knowledge base for joint intent detection and slot filling that was proposed in this paper is to identify the intent and slot of the text that a user inputs in the dialog system. In this paper, the accuracy is used to evaluate the intent-detection task, and the F value is used to evaluate the slot-filling task. The experimental results are

shown in Table 2. With the ATIS dataset, the accuracy values of intent detection and the F values of slot filling obtained by our model are 98.83% and 97.06% respectively, and with the Snips dataset, the accuracy value and F value are 98.79% and 97.31%, respectively. For comparison, the RNN-LSTM [26], attention-based RNN [27], slot-gated [28], joint BERT [33], and CAPSULE-NLU [37] models have demonstrated their performance on the ATIS and Snips datasets. It can be seen from Table 2 that, according to the accuracy index, our model outperforms the best joint BERT model [33] by 1.33% and 0.19% among the group of comparative models. Meanwhile, with respect to the F index, our model outperforms the best joint BERT [33] by 0.94% and 0.31% among the group of comparative models. As shown in Table 2, we performed ablation analysis on the ATIS dataset. In the ATIS dataset, without joint learning, the accuracy of intent detection drops to 96.19% from 97.40%, and the F1 score of slot filling drops to 94.90% from 96.16%. Without a knowledge base, the accuracy of intent detection drops to 98.37% from 98.83%, and the F1 score of slot filling drops to 96.42% from 97.06%. The Snips dataset demonstrates the same trends. In other words, strong associations and mutual promotion between tasks may improve the intent-detection and slot-filling performance. Moreover, integrating knowledge base facilitates the performance improvement of intent-detection and slot-filling modules.

Table 2. Performance comparison.

Model	ATIS		Snips	
	Intent Accuracy (%)	F1 Score (%)	Intent Accuracy (%)	F1 Score (%)
RNN-LSTM [26]	94.09	95.42	96.90	87.30
Joint BERT [33]	97.50	96.10	98.60	97.00
Attention-based RNN [27]	91.10	94.20	96.70	87.80
Slot-gated [28]	94.10	95.20	97.00	88.80
CAPSULE-NLU [37]	95.00	95.20	97.30	91.80
MTL (One-hot)	97.40	96.16	97.66	92.32
MTL (BERT)	98.37	96.42	98.52	97.03
MTL with knowledge base	98.83	97.06	98.79	97.31

Three reasons may contribute to the superiority of our joint model, namely, the shared parameters, knowledge base, and joint promotion. First, our joint model uses the LSTM–CNN shared representation layer to obtain shared resources, which can be used to better learn more common information from the dataset and can lead to high scalability. Second, a knowledge base is introduced into the model to improve its performance. Third, we designed a weighted-loss function to build the joint optimization of these modules. In other words, strong associations and external knowledge bases between tasks promote the model's performance.

Here, we discuss an error analysis of the proposed methods. This error analysis will help improve our model. Two factors lead to error in our model. The first is lexical ambiguity. Ambiguity is a natural and common linguistic phenomenon in English. Some nouns were incorrectly identified as verbs, such as in the following sentence: “I would give this current book a rating of five and a best rating of six.” If we consider the word “book” as a verb, it should be marked as “O”. However, this word is a noun and should be marked as “object_type”. The second factor is the entity recognition problem. Some phrases cannot be recognized as an entity; e.g., “Find a photograph called ‘call on me’.” Here, “call on me” is an entity but cannot be recognized, which means we could not find the correct referred entity for the given mention.

5. Discussion

In this section, we demonstrate the influencing factors in our joint model, namely, shared representation, the BERT coding method, the knowledge base, and the learning-weighted loss function, respectively. First, we compared our MTL model based on the one-hot coding method with its pruning version that executes independent tasks on the

ATIS dataset. The experimental results in Table 3 show that our joint model substantially outperforms both independent intent-detection and slot-filling models.

Table 3. Comparison of independent and joint models.

	Intent Accuracy (%)	F1 Score (%)
Independent training intent	96.19	-
Independent training slot	-	94.90
Joint model	97.40	96.16

Then, we analyzed the effects of the BERT pre-train language model on this model. BERT is helpful for textual encoding and suitably manages tasks with deep semantic features. In general, applying this method can greatly enhance the performance of the NLP tasks with just a few words. Compared with the one-hot encoding, the effectiveness increased by 0.97% and 0.26% separately.

Furthermore, we further analyzed the key to the success of our joint model. From the shared resources standpoint, the shared parameters and features have been learned to represent the high correlation between intents and slots, which can fully determine the relationship between these two tasks.

We introduced the WordNet knowledge base as an external knowledge source to the Bi-LSTM model. The experimental results demonstrate that this method solves the problem of unknown words, and the model performance improves. This is because some knowledge existing in the knowledge base can help identify words that do not appear in the dataset. Furthermore, we selected a case from ATIS. Here, for example, if the dataset contains only the word “American”, the phrase “the United States” cannot be identified by relying solely on this dataset. However, if a knowledge base is introduced, by determining that “America” and “the United States” are synonyms, the phrase “the United States” can be identified.

In addition, we analyze the mechanism of the weighted-loss function in our joint model, which simultaneously affects the intent-detection and the slot-filling task. We define α as the weight coefficient, and the value of α obtains the optimal configuration by using the gradient descent algorithm to solve the problem that the fixed-value weights cannot be reasonably distributed due to subjective factors. Furthermore, we show the influence of the self-learning weight based on the gradient descent algorithm. Figure 3 illustrates the change of α in a certain iteration, which indicates that the weight can be acquired through learning.

Figure 3. Self-learning weight α value changes.

The value of α continuously changes as the number of iterations increases. The value of α is used for making the weighted-loss function converge faster than the original loss function. Figure 4 shows the comparison of the changes of the weighted-loss function,

loss1, with the changes of the original loss function, loss2. This picture indicates that the weighted-loss function converges faster.

Figure 4. Loss function change.

The experiments show that with the weighted self-learning method, the intent-detection accuracy is 98.83%, which is a 0.2 percentage point increase compared to the former subjective weight determination.

6. Conclusions

In order to completely utilize the incidence relations and shared resources between the two modules by only relying on present joint models, and exploring the value of knowledge base to these modules, this paper proposes a joint model for intent detection and slot filling based on MTL with a knowledge base, which makes full use of the external knowledge, and a high-quality relationship information between intents and slots. Firstly, we obtained the shared parameters and features between two modules based on the neural networks of LSTM and CNN. Secondly, the knowledge base was introduced into the model to improve its performance. Finally, a weighted-loss function was built to optimize the whole joint model. Experiments are based on the ATIS and Snips dataset to evaluate the performance of the proposed method. The experimental results show that the accuracy of the intent detection was 98.83% and the F value of slot filling was 97.06%, both on the ATIS dataset. On the Snips dataset, the accuracy of the intent detection and the F value of slot filling are 98.79% and 97.31% respectively. Furthermore, we also analyzed the keys to the success of our joint model, i.e., sharing representation, the BERT coding method, the knowledge base and learning-weighted-loss function, which can fully determine the relationship between these two tasks and external knowledge. The results demonstrate that our model is feasible and superior to the related baseline model. Certainly, the proposed method still has some limitations. For example, our method is evaluated on the small-scale datasets, which may limit the number of observation and behavioral characteristics. In the future researches, the method proposed in this paper can be extended to more complex and diversified datasets, or trying something new with dialogue systems.

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

Funding: This research was funded by the China National Social Science Fund (19BXW110).

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. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Hakkani-Tur, D.; Ju, Y.; Zweig, G.; Tur, G. Clustering Novel Intents in a Conversational Interaction System with Semantic Parsing. In Proceedings of the Interspeech, Dresden, Germany, 6–10 September 2015; pp. 1854–1858.
2. Tur, G.; Deng, L. Intent determination and spoken utterance classification. In *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech*; Wiley: Hoboken, NJ, USA, 2011; pp. 93–118.
3. Celikyilmaz, A.; Hakkani-Tur, D.; Tur, G. Statistical semantic interpretation modeling for spoken language understanding with enriched semantic features. In Proceedings of the 2012 IEEE Spoken Language Technology Workshop (SLT), Miami, FL, USA, 2–5 December 2012; pp. 216–221.
4. Henderson, J.; Jurčiček, F. *Data-Driven Methods for Spoken Language Understanding*; Springer: New York, NY, USA, 2012; pp. 19–38.
5. Liu, J.; Cyphers, S.; Pasupat, P. A conversational movie search system based on conditional random fields. In Proceedings of the 13th Annual Conference of the International Speech Communication Association, Portland, OR, USA, 9–13 September 2012; pp. 1–4.
6. Wei, Y.; Zhang, L.; Zhang, Y.; He, L.; Fang, D. Combining support vector machines, border revised rules and transformation-based error-driven learning for Chinese chunking. In Proceedings of the 2010 International Conf. Artificial Intelligence and Computational Intelligence, Sanya, China, 23–24 October 2010; pp. 383–387.
7. Hashimoto, K.; Xiong, C.; Tsuruoka, Y.; Socher, R. A joint many-task model: Growing a neural network for multiple NLP tasks. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 7–11 September 2017; pp. 446–451.
8. Gokhan, T.; Heck, L.; Parthasarathy, S. Sentence simplification for spoken language understanding. In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal, Prague, Czech Republic, 22–27 May 2011; pp. 5628–5631.
9. Francois, M.; Mairesse, F.; Gasic, M.; Jurcicek, F.; Keizer, S.; Thomson, B.; Yu, K.; Young, S. Spoken language understanding from unaligned data using discriminative classification models. In Proceedings of the 2009 IEEE International Conference on Acoustics, Speech and Signal, Taipei, Taiwan, 19–24 April 2009; pp. 4749–4752.
10. Jeong, M.; Geunbae, L. Triangular-chain conditional random fields. *IEEE Trans. Audio Speech Lang. Process.* **2008**, *16*, 1287–1302. [[CrossRef](#)]
11. Yao, K.; Yao, K.; Peng, B.; Zweig, G.; Yu, D.; Li, X.; Gao, F. Recurrent conditional random field for language understanding. In Proceedings of the 2014 IEEE International Conference on Acoustics, Speech and Signal Processing, Florence, Italy, 4–9 May 2014; pp. 4077–4081.
12. Min, Y.; Shin, Y.; Lee, S. Data augmentation for spoken language understanding via joint variational generation. In Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; pp. 7402–7409.
13. Xu, M.; Xu, M.; Yang, R.; Ranshous, S.; Li, S.; Samatova, N.F. Leveraging external knowledge for phrase-based topic modeling. In Proceedings of the 2017 Conference on Technologies and Applications of Artificial Intelligence, Taipei, Taiwan, 1–3 December 2017; pp. 29–32.
14. Firdaus, M.; Kumar, A.; Ekbal, A.; Bhattacharyya, P. A multi-task hierarchical approach for intent detection and slot filling. *Knowl. Based Syst.* **2019**, *183*. [[CrossRef](#)]
15. Augenstein, I.; Søgaard, A. Multi-task learning of keyphrase boundary classification. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, BC, Canada, 30 July–4 August 2017; pp. 232–242.
16. Xiong, C.; Zhong, V.; Socher, R. Dynamic coattention networks for question answering. In Proceedings of the 6th International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018; pp. 1–14.
17. Sarikaya, R.; Hinton, G.E.; Ramabhadran, B. Deep belief nets for natural language call-routing. In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, Prague, Czech Republic, 22–27 May 2011; pp. 5680–5683.
18. Celikyilmaz, A.; Celikyilmaz, A.; Hakkani-tur, D.; Tur, G.; Fidler, A.; Hillard, D. Exploiting distance based similarity in topic models for user intent detection. In Proceedings of the 2011 IEEE Workshop on Automatic Speech Recognition & Understanding, Waikoloa, HI, USA, 11–15 December 2011; pp. 425–430.
19. Ji, Y.; Hakkani-Tür, D.; Celikyilmaz, A.; Heck, L.; Tur, G. A variational bayesian model for user intent detection. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Florence, Italy, 4–9 May 2014; pp. 4072–4076.
20. Troussas, C.; Krouska, A.; Sgouropoulou, C.; Voyiatzis, I. Ensemble learning using fuzzy weights to improve learning style identification for adapted instructional routines. *Entropy* **2020**, *22*, 735. [[CrossRef](#)] [[PubMed](#)]
21. Giannakas, F.; Troussas, C.; Voyiatzis, I.; Sgouropoulou, C. A deep learning classification framework for early prediction of team-based academic performance. *Appl. Soft Comput.* **2021**, *106*. [[CrossRef](#)]

22. Mesnil, G.; Dauphin, Y.; Yao, K.; Bengio, Y.; Deng, L.; Hakkani-Tur, D.; He, X.; Heck, L.; Tur, G.; Yu, D.; et al. Using recurrent neural networks for slot filling in spoken language understanding. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2015**, *23*, 530–539. [\[CrossRef\]](#)
23. Lin, R. Combining word feature vector method with the convolutional neural network for slot filling in spoken language Understanding. *arXiv* **2018**, arXiv:1806.06874.
24. Xu, Z.; Che, W.; Liu, T. Slot filling based on Bi-LSTM-CRF. *Intell. Comput. Appl.* **2017**, *6*, 94–97.
25. Xu, P.; Sarikaya, R. Convolutional neural network based triangular crf for joint intent detection and slot filling. In Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding, Olomouc, Czech Republic, 8–12 December 2013; pp. 78–83.
26. Dilek, H. Multi-Domain joint semantic frame parsing using Bi-Directional RNN-LSTM. In Proceedings of the Interspeech, San Francisco, CA, USA, 8–12 September 2016; pp. 715–719.
27. Liu, B.; Ian, L. Recurrent neural network structured output prediction for spoken language understanding. In Proceedings of the NIPS Workshop on Machine Learning for Spoken Language Understanding and Interactions, Montreal, QC, Canada, 11 December 2015; pp. 1–7.
28. Hua, B.; Yuan, Z.; Xiao, W. Joint Slot Filling and Intent Detection with BLSTM-CNN-CRF. *Comput. Eng. Appl.* **2018**, *6*, 1–7.
29. Coo, C.; Gao, G.; Hsu, Y.; Huo, C.; Chen, T.; Hsu, K.; Chen, Y. 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, New Orleans, LA, USA, 1–6 June 2018; pp. 753–757.
30. 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, Brussels, Belgium, 31 October–4 November 2018; pp. 3824–3833.
31. Haihong, E.; Niu, P.; Chen, Z.; Song, M. A novel bi-directional interrelated model for joint intent detection and slot filling. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 5467–5471.
32. Chen, M.; Zeng, J.; Lou, J. A self-attention joint model for spoken language understanding in situational dialog applications. *arXiv* **2019**, arXiv:1905.11393.
33. Chen, Q.; Zhuo, Z.; Wang, W. BERT for joint intent classification and slot filling. *arXiv* **2019**, arXiv:1902.10909.
34. Zhang, Z.; Zhang, Z.; Zhang, Z.; Chen, H.; Zhang, Z. A joint learning framework with BERT for spoken language understanding. *IEEE Access* **2019**, *7*, 168849–168858. [\[CrossRef\]](#)
35. Castellucci, G.; Bellomaria, V.; Favalli, A.; Romagnoli, R. Multi-lingual intent detection and slot filling in a joint BERT-based Model. *arXiv* **2019**, arXiv:1907.02884.
36. Price, P.J. Evaluation of spoken language systems: The ATIS domain. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, PA, USA, 24–27 July 1990, Proceedings of the Workshop Held at Hidden Valley, PA, USA, 24–27 July 1990*; Morgan Kaufmann Publications: Burlington, MA, USA, 1990; pp. 91–95.
37. 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.