Jointly Modeling Intent Identification and Slot Filling with Contextual and Hierarchical Information

Liyun Wen*, Xiaojie Wang*, Zhenjiang Dong** and Hong Chen**

*Beijing University of Posts and Telecommunications
**ZTE Corporation

{wenliyun,xjwang}@bupt.edu.cn
dongzhenjiangvip@163.com
chen.hong3@zte.com.cn

Abstract. Intent classification and slot filling are two critical subtasks of natural language understanding (NLU) in task-oriented dialogue systems. Previous work has made use of either hierarchical or contextual information when jointly modeling intent classification and slot filling, proving that either of them is helpful for joint models. This paper proposes a cluster of joint models to encode both types of information at the same time. Experimental results on different datasets show that the proposed models outperform joint models without either hierarchical or contextual information. Besides, finding the balance between two loss functions of two subtasks is important to achieve best overall performances.

Keywords: Joint Models · Contextual and Hierarchical Information · Natural Language Understanding

1 Introduction

Natural Language Understanding (NLU), which refers to the targeted understanding of human language directed at machines [1], is a critical component in dialogue systems. An NLU system typically consists of three subtasks, namely domain identification, intent classification and slot filling [2].

Conventionally, the subtasks are processed in a pipeline framework; firstly the domain of an input is detected, secondly the intent is classified and finally the semantic slots are extracted. Lots of work has been done for each subtask, respectively. For example, Haffner et al. (2003) [3] built a Support Vector Machines (SVM) based classifier to classify intent (call) labels. Yao et al. (2014) [4] investigated Long Short-Term Memory (LSTM) methods for slot filling. Pipeline systems not only suffer from the problem of error accumulation, but also cannot model the interaction between different subtasks.

Recent work has shown the advantages of jointly modeling NLU subtasks. The ability of featuring the correlations between subtasks helps joint models achieve competitive performances. Shi et al. (2015) [5] proposed a Recurrent Neural Network (RNN) model to jointly optimize domain identification, intent classification and slot filling, which obtained state-of-the-art results on ATIS dataset. Hakkani et al. (2016) [6] presented a method for simultaneously modeling domain recognition, intent classification

and slot filling by introducing an extra token <EOS> for sentence-level labels in an LSTM-based slot sequential labeling model.

Apart from correlative information, hierarchical structure is considered as another useful information for joint modeling. Zhou et al. (2016) [7] proposed a hierarchical Long Short-Term Memory (HLSTM) model to implement intent identification in the lower layer and slot filling in the higher layer. They demonstrated joint models with hierarchical structure outperformed non-hierarchical joint methods. But contextual information was not used in their model. Previous work on single subtask has proved that contextual information is an effective feature for NLU subtasks. Yao et al. (2013) [8] presented a window-based RNN model to capture contextual features in NLU. Mesnil et al. (2015) [9] applied bidirectional Elman and Jordan RNN to encode the future and past information in inputs during slot filling.

We think that contextual and hierarchical information help NLU subtasks in different dimensions. Hierarchy could characterize the nature order among different tasks: as pointed in [10], primary tasks are better kept at the lower layers in a deep network. While contextual idiosyncrasies could bring the richness of representation by observing features from preceding and following positions in its vicinity, which could facilitate a morpheme/word unit based recognition task like slot filling. It is therefore possible to improve performances by combining both of them. Zhang et al. (2016) [11] proposed a two-layer hierarchical joint model, with a lower RNN tackling slot filling and an upper max-pooling handling intent recognition. Liu et al. (2016) [12] presented an attention-based RNN model with both contextual and hierarchical information captured, which obtained better results on ATIS dataset.

In order to encode the two kinds of information and detail the specific effects of them, this paper proposes a cluster of contextual hierarchical joint (CHJ) models to jointly model intent classification and slot filling. The models have a two-layer-LSTM structure, where intent classification and slot filling are dealt by different layers. Distinguished from HLSTM, our proposed models take bi-directional or backward order to utilize contextual information. All parameters are learned simultaneously to minimize a joint loss function, i.e. the weighted sum of two losses. Experiments show that on different NLU datasets CHJ models outperform non-hierarchical or non-contextual models, respectively.

The rest of the paper is structured as follows. Sect.2 demonstrates our proposed models in detail; Sect.3 presents the tasks and experimental results; and finally, conclusions are drawn in Sect.4.

2 Models

LSTM is the basic unit in all CHJ models. We therefore introduce LSTM first, and then propose our models.

2.1 LSTM

LSTM [13], a variant of RNN, consists of one or more memory cells and three nonlinear summation units, i.e. the input, output and forget gate. Detailed introduction and equa-

tions about LSTMs can be found in [14]. At time t, the calculating process of hidden state vector $\boldsymbol{h}_t$ is abbreviated as follows:

$$\boldsymbol{h}_t = LSTM(\boldsymbol{W}_x \boldsymbol{x}_t + \boldsymbol{W}_h \boldsymbol{h}_{t-1}) \tag{1}$$

where $LSTM$ is the recurrent neural function that calculates the current hidden state vector $\boldsymbol{h}_t$ given the previous one $\boldsymbol{h}_{t-1}$ and the current input $\boldsymbol{x}_t$. $\boldsymbol{W}_x$ and $\boldsymbol{W}_h$ are the associated weight matrices.

According to the processing order, LSTMs can be classified to different categories: forward LSTMs, backward LSTMs and bi-directional LSTMs. Forward LSTMs take the standard forward order when reading sequences, and symmetrically backward LSTMs read the sequence in a reversed way. Bi-directional LSTMs (bi-LSTMs) [17] present each sequence forwards and backwards to two separate LSTM hidden layers and concatenate both to the same output layer [14].

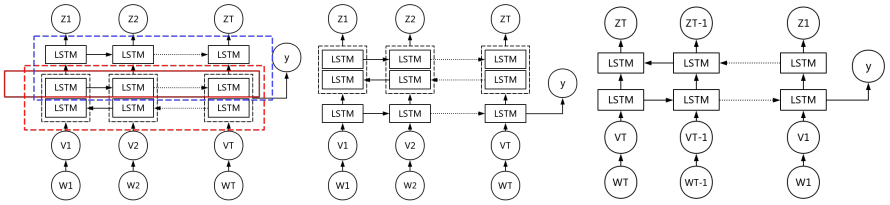
2.2 Contextual Hierarchical Joint (CHJ) Models

Given an input utterance $w_{(1:T)} = (w_1, w_2, ..., w_T)$, NLU is to predict an intent class for current utterance and to label slot tags among all words. Let $Y = \{Y_1, Y_2, ..., Y_M\}$ denote the intent label set and $S = \{S_1, S_2, ..., S_N\}$ denote the slot set. Slot filling is implemented via sequence labeling methods. All slot classes are transformed into semantic tags according to the IOB annotated method [18]; each slot class S_i can generate two semantic tags: $B-S_i$ and $I-S_i$, as well as the label O representing the out-of-slot tag. The corresponding semantic tags set can be denoted as $Z = \{Z_1, Z_2, ..., Z_{2N+1}\}$. Thus the process is to map $w_{(1:T)} = (w_1, w_2, ..., w_T)$ to a predicted intent label y and a set of semantic tags $z_{(1:T)} = (z_1, z_2, ..., z_T)$.

A hierarchical LSTM is built to model the mapping at first. The structure of the model is shown in Fig.1(a). It is a two-layer LSTM, where a forward LSTM is stacked on the top of a bi-LSTM. The overall flow of information is from the lower to the upper, namely the upper layer takes the hidden state vector of the lower layer directly as input, and inputs of the lower bi-LSTM are the embedding vectors of words in the current sentence. Intent classification is tackled by the lower layer: the final output of the bi-LSTM is fed to a softmax classifier to get an intent label of the sentence. Slot filling is dealt by the upper: the output of each LSTM unit is fed to a softmax classifier to get a slot label of the corresponding word.

This structure tries to utilize both hierarchical and contextual information in jointly modeling intent identification and slot filling. The hierarchical structure is used to capture internal relations, like order or dependency, of two subtasks. The bi-LSTM is used to capture contextual idiosyncrasies from past and future positions of a certain word during the current sentence.

For an input sentence, each word $w_t (t = 1, 2, ..., T)$ is first mapped into its embedding vector $\boldsymbol{v}_t (t = 1, 2, ..., T)$. Bi-directionally taking the embedding representations as input, according to (1), the lower layer calculates two sets of hidden state vectors and concatenate them into one: $\boldsymbol{h}^1_{(1:T)} = (\boldsymbol{h}^1_1, ..., \boldsymbol{h}^1_T)$. The last hidden state vector $\boldsymbol{h}^1_T$ is fed into a softmax classifier. The probability distribution of all intent labels $\boldsymbol{y}$ is obtained by softmax.



(a) CHJ(i.bi_s.fw) (b) CHJ(i.fw_s.bi) (c) CHJ(i.bw_s.fw)

Fig. 1. Some structures of the cluster of CHJ models

$$\boldsymbol{y} = softmax(\boldsymbol{W}^1 \boldsymbol{h}_T^1), \tag{2}$$

where $\boldsymbol{W}^1$ is the softmax weight matrix of the lower LSTM. A predicted intent tag can be calculated by getting argmax of $\boldsymbol{y}$.

The upper layer takes the hidden state vector of the lower layer as input directly.

$$\boldsymbol{x}_t^2 = \boldsymbol{h}_t^1, t = 1, 2, ..., T \tag{3}$$

where $\boldsymbol{x}_t^2$ denote the upper layer inputs.

Following (1) and (3), we get the set of hidden state vectors of the upper layer $\boldsymbol{h}_{(1:T)}^2 = (\boldsymbol{h}_1^2, ..., \boldsymbol{h}_T^2)$. Every hidden state vector is fed into a softmax to obtain corresponding probabilities $\boldsymbol{z}_t$.

$$\boldsymbol{z}_t = softmax(\boldsymbol{W}^2 \boldsymbol{h}_t^2), t = 1, 2, ..., T \tag{4}$$

where $\boldsymbol{W}^2$ is the softmax weight matrix of the upper LSTM. Argmax can be used on $\boldsymbol{z}_t$ to get predicted slot tags.

The parameter set of the whole network is $\theta = \{\boldsymbol{W}_x, \boldsymbol{W}_h, \boldsymbol{W}^1, \boldsymbol{W}^2\}$. All parameters of two tasks are learned simultaneously to minimize a joint objective function $J(\theta)$, which is represented as the weighted sum of two losses, together with an l_2-norm term:

$$J(\theta) = \alpha L_I + (1 - \alpha) L_S + \frac{\lambda}{2} ||\theta||_2^2, 0 \leq \alpha \leq 1, \tag{5}$$

where L_S represents the slot filling loss and L_I represents the intent identification loss. Let D be the whole training set and $L(\cdot)$ be the cross-entropy operation. Suppose $\hat{\boldsymbol{y}}^{(i)}$ and $\hat{\boldsymbol{z}}_{(1:T)}^{(i)}$ are the true intent label and semantic tags of the i^{th} training sample. The two losses are calculated as follows:

$$L_S = \frac{1}{|\mathcal{D}|} \sum_i^{|\mathcal{D}|} \frac{1}{T} \sum_{t=1}^{T} L(\boldsymbol{z}_t^{(i)}, \hat{\boldsymbol{z}}_t^{(i)}), \tag{6}$$

$$L_I = \frac{1}{|\mathcal{D}|} \sum_i^{|\mathcal{D}|} L(\boldsymbol{y}^{(i)}, \hat{\boldsymbol{y}}^{(i)}), \tag{7}$$

The tradeoff between two objectives relies on the hyper-parameter α. When α is bigger than 0.5, the total joint loss function pays more attention to the intent identification. On the contrary, when α is smaller than 0.5, the slot loss plays a more important role in supervised learning.

The model can be thought as an extension of several previous models by combing contextual or hierarchical information. If an intent-relevant tag is attached before or after each sentence, sentence-level intent identification and word-level slot filling can be jointly modeled in a single sequence labeling model. The forward LSTM in the red solid frame in Fig.1(a) is one kind of these sequence labeling models by feeding each unit's output to a softmax classifier, as proposed in [6] (forward Hakkani's Model, denoted as Fw-Hakkani's Model afterwards). It is a flap style of joint models, only correlative information between two subtasks is taken into consideration in this joint model. Based on the flap structure, two improvements can be made. One is shown in the red dotted frame, in which contextual information is included by using bi-LSTM (denoted as Bi-Hakkani's Model). The other is highlighted by the blue dotted frame, where the flap structure was improved to become a hierarchical structure as Zhou et al. proposed [7] (denoted as Zhou's Model). A two-layer LSTM (not bi-LSTM) was used in Zhou's model, where the upper layer is for slot filling and the lower layer is for intent identification. It is obvious that structures in red dotted frame and blue dotted frame extend Fw-Hakkani's Model in two different dimensions. Our model combines both of them.

The proposed model can have several different variants by changing the way of combining contextual and hierarchical information. We consider two different ways to include contextual information for slot filling. One is to use bi-LSTM for slot filling no matter the subtask is modeled in the lower layer or the upper; another way is to tackle slot filling by an LSTM in the upper layer, and an LSTM with the inversed direction is employed in the lower layer for intent identification. By using two inversed LSTMs instead of one LSTM and one bi-LSTM, the model can be simplified. We also consider two possible hierarchies: one is putting intent identification in the lower and slot filling in the upper, the other is an interchange of layers in the precious one.

Table 1. Several variants of model CHJ(i.bi_s.fw)

Denotation	Description
CHJ(i.fw_s.bi)	A forward LSTM for intent classification in the lower layer; the upper layer tackles slot filling by a bi-LSTM (Fig.1(b)).
CHJ(i.bw_s.fw)	A backward LSTM for intent classification in the lower layer; a forward LSTM for slot filling in the upper layer (Fig.1(c)).
CHJ(i.fw_s.bw)	A forward LSTM for intent classification in the lower layer; a backward LSTM for slot filling in the upper layer.
CHJ(s.bi_i.fw)	The lower layer deals with slot filling by a bi-LSTM; while the upper layer solves intent classification forwards.

For convenience, the model illustrated in Fig.1(a) is denoted by CHJ(i.bi_s.fw): i.bi before the underscore describes the lower layer structure, representing a bi-LSTM model for intent identification; s.fw after the underscore describes the upper layer structure, denoting a forward LSTM structure for slot filling. By using these denotation, we elucidate several variants of model CHJ(i.bi_s.fw) in Table1.

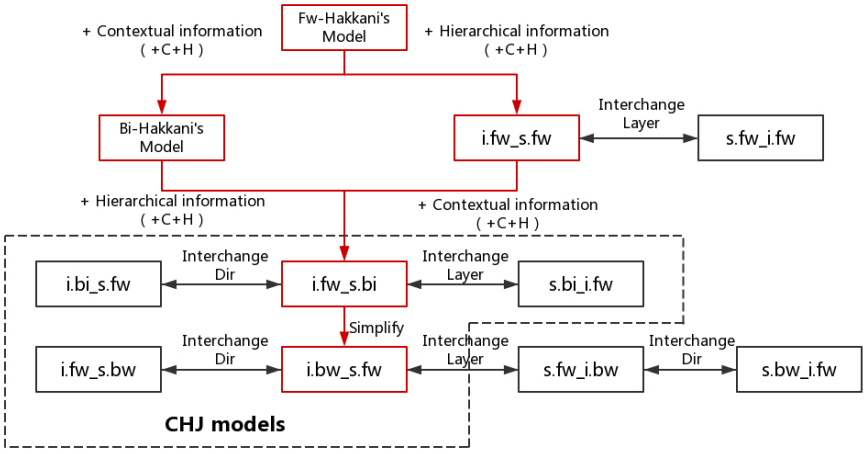


Fig. 2. The cluster of proposed CHJ models and their transformation relationship

The transformative relations between these models are illustrated in Fig.2, where "Simplify" represents the operation that utilizes reversed directions to replace bi-LSTM, as we pointed before; "InterchangeLayer" denotes the operation that totally interchanges the two layers, such as the operation between i.fw_s.bi and s.bi_i.fw; "InterchangeDir" represents the operation that remains the tasks order unchanged and interchanges the directions of two layers.

In Fig.2, from Fw-Hakkani's Model, there are two paths to add hierarchical and contextual information. One is to add contextual information first and then the hierarchical one, as shown in the left path; the other is an inverse order, which is listed in the right path. Both paths lead to the same destination i.fw_s.bi, which can be simplified into i.bw_s.fw. Based on i.fw_s.bi and i.bw_s.fw, certain operations can be implemented to generate some other CHJ models. All models in the dotted frame are CHJ models. We should note that s.fw_i.bw and s.bw_i.fw are not CHJ models. Both of them condition slot filling in the lower layer; no matter how they change the lower direction (forwards or backwards), slot filling cannot simultaneously take both past and future context into consideration. More precisely, simplified versions of contextually modeling slot filling, viewed as substitutes for bi-LSTMs, need slot filling to be tackled by the upper layer.

3 Experiments

First, the datasets and the experimental setup are illustrated. Second, our results and related benchmarks are compared. Then the tradeoff between multi-tasks is discussed in detail. Finally, the case study is presented.

3.1 Datasets and Settings

The experiments are implemented in three different corpora: the DSTC2[1] [19], DSTC5[2] [20] and our Chinese meeting room reservation corpus collected from a Chinese meeting-room reservation system (CMRS). Basic information about these three corpora is listed in Table2.

Table 2. The number of sentences in each corpus

Dataset	Train	Dev	Test
DSTC2	4,790	1,579	4,485
DSTC5	27,528	3,441	3,447
CMRS	2,901	969	967

In DSTC2, each user utterance with only one intent (act) label is used. The number of intent labels is 13, the number of different slots is 4, and thus the total number of semantic tags is 9 (2*4+1; each slot can generate two semantic tags: $B-slot$ and $I-slot$, as well as the label O).

In DSTC5, each user utterance with only one intent (act) label is used. The number of intent labels is 84, the number of different fine-grained slots is 266, and thus the total number of semantic tags is 533. In order to exclude the influence of cross-language problem, only English sentences are used.

As for CMRS, the number of intent labels is 5, the number of different slots is 5, and thus the total number of semantic tags is 11.

We choose commonly used configurations for experimental settings. For each group of tasks, we use AdaGrad [21] with mini-batches [22] to minimize the objective function. Derivatives are calculated from standard back-propagation. The model achieving the best performance on the development set is used as the final model to be evaluated. Statistical significance tests are implemented by 5-fold cross validation and Student's t-test, with the significance level set to 0.05.

3.2 Comparisons with Recent Work

Table3, Table4 and Table5 exhibit the experimental results on DSTC2, DSTC5 and CMRS, respectively. Performances are computed in terms of slot F1-measure (at the

[1] http://camdial.org/~mh521/dstc/

[2] http://workshop.colips.org/dstc5/

Table 3. Results on DSTC2 corpus

Information (C/H)	Model	Slot(F1)%	Intent(F1)%	Avg(F1)%
None	Fw-Hakkani's Model	96.22	99.22	97.72
+C	Bi-Hakkani's Model	98.92	99.18	99.05
+H	i.fw_s.fw	96.67	99.23	97.95
	s.fw_i.fw	96.64	99.46	98.05
	s.fw_i.bw	96.62	99.39	98.01
	s.bw_i.fw	97.96	99.48	98.72
+C+H (CHJ models)	s.bi_i.fw	98.96	99.45	99.21
	i.fw_s.bi	98.81	99.33	99.07
	i.bi_s.fw	98.93	99.37	99.15
	i.bw_s.fw	**99.01**	**99.48**	**99.25**
	i.fw_s.bw	98.91	99.34	99.13

Table 4. Results on DSTC5 corpus

Information (C/H)	Model	Slot(F1)%	Intent(F1)%	Avg(F1)%
None	Fw-Hakkani's Model	36.22	53.02	44.62
+C	Bi-Hakkani's Model	45.17	52.54	48.86
+H	i.fw_s.fw	23.45	49.15	36.30
	s.fw_i.fw	33.45	**53.48**	43.47
	s.fw_i.bw	35.99	52.89	44.44
	s.bw_i.fw	38.81	53.27	46.04
+C+H (CHJ models)	s.bi_i.fw	**45.51**	53.35	**49.43**
	i.fw_s.bi	40.29	51.84	46.07
	i.bi_s.fw	39.36	50.84	45.10
	i.bw_s.fw	42.85	51.38	47.12
	i.fw_s.bw	35.68	49.65	42.67

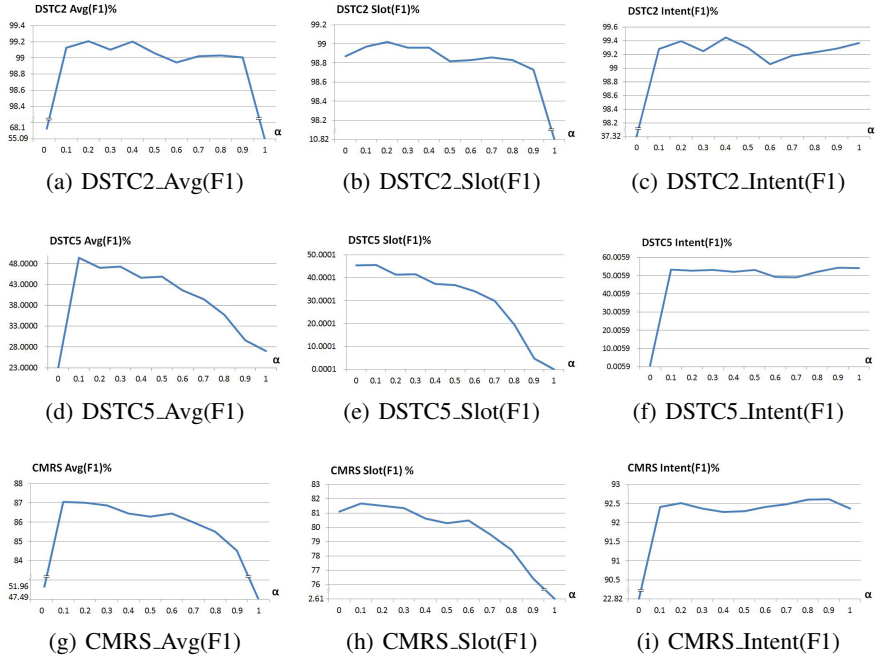
Table 5. Results on CMRS corpus

Information (C/H)	Model	Slot(F1)%	Intent(F1)%	Avg(F1)%
None	Fw-Hakkani's Model	61.16	92.61	76.89
+C	Bi-Hakkani's Model	82.91	92.69	87.80
+H	i.fw_s.fw	61.83	**92.84**	77.34
	s.fw_i.fw	61.15	92.58	76.87
	s.fw_i.bw	61.72	92.72	77.22
	s.bw_i.fw	67.61	92.44	80.03
+C+H (CHJ models)	s.bi_i.fw	81.68	92.41	87.05
	i.fw_s.bi	84.14	92.76	88.45
	i.bi_s.fw	84.93	92.53	88.73
	i.bw_s.fw	**85.80**	92.60	**89.20**
	i.fw_s.bw	83.91	92.75	88.34

slot level)[3], intent F1-measure (at the label level) and the average of both, as a measure of overall performance.

From the three tables, we have some general conclusions. 1) The model that gets best overall performance is always among the CHJ models on all three datasets, proving the effectiveness of combining hierarchical and contextual information in joint modeling of NLU. 2) Once hierarchical information is included, an improvement on intent identification is achieved, which suggests that hierarchical information is helpful, especially for intent classification. (The improvements are statistical significant; from None to +H: 4.0836>2.7764; from +C to +C+H: 5.9496>2.7764) 3) If we introduce contextual information into models, the slot performance gets a considerable boost; this phenomenon indicates that contextual information could benefit slot filling to a great extent. (The improvements are statistical significant; from None to +C: 41.8951>2.7764; from +H to +C+H: 12.5448>2.7764) 4) Among the four +H models, s.bw_i.fw performs better than any other one does in slot filling task, indicating that backward encoding is more helpful than forward encoding. We notice that slot values often consist of phrases in the form of pre-modifications + head words; in this structure, the posteriori head words play a vital role in slot recognition. Backward networks could previously see the posteriori center words and therefore perform better than forward networks which have troubles in labeling those pre-modifiers without the information of head words.

[3] http://www.cnts.ua.ac.be/conll2000/chunking/conlleval.txt



(a) DSTC2_Avg(F1) (b) DSTC2_Slot(F1) (c) DSTC2_Intent(F1)

(d) DSTC5_Avg(F1) (e) DSTC5_Slot(F1) (f) DSTC5_Intent(F1)

(g) CMRS_Avg(F1) (h) CMRS_Slot(F1) (i) CMRS_Intent(F1)

Fig. 3. α-performances curves on three datasets

For structures in the form of head word + post-modification, forward networks could perform better. Transparently, contextual (such as bi-directional) networks do the best. 5) All corpora support that i.bw_s.fw gets higher performances than i.bi_s.fw does. As we pointed in Sect.2.2, i.bw_s.fw is a simplified version of i.bi_s.fw. They combine contextual and hierarchical information in similar ways, but i.bw_s.fw has a more concise structure, which conduces to better performances.

3.3 Tradeoff between Multi-tasks

α is the parameter used to leverage the loss functions of two tasks in CHJ models. When α is bigger than 0.5, the total joint loss function pays more attention to the intent loss. On the contrary, when α is smaller than 0.5, the slot loss plays a more important role in supervision.

Figure3 shows the α-performances curves of model CHJ(s.bi_i.fw) on different datasets. By summarizing the universality, we can draw several points. 1) Slot filling, intent classification and overall performance all get best results when $\alpha \neq 0$ and $\alpha \neq 1$, supporting that the interaction of two tasks can be beneficial if they are properly combined. 2) More specifically, a smaller non-zero α could help achieve more competitive overall performance. α=0.1 or α=0.2 seems to be a good choice. 3) It is clear that for slot filling, a small α has absolute advantages compared with a larger one. 4) For intent classification, small α and large α bring comparable results. It can be referred that two losses are equally important for intent recognition in joint modeling of NLU.

3.4 Case Study

An example in CMRS corpus is listed in Table6, where ”30 号” (thirty date) is a Chinese time expression. ”30” can be used in both time-relevant and number-relevant slots. In this case, contextual information is demanded for disambiguating. It can be seen that models with only hierarchical (+H) information mislabel ”30”. While, models with only contextual information label ”30” correctly, but misjudge the intent label. In fact, a time slot is a strong hint for intent identification. It seems hierarchical structures can make better use of this hint. CHJ models, which combine the two kinds of information, do a correct work on both subtasks.

Although CHJ models have taken considerable results, there are still some places to be improved. Table7 shows two transcripts in which CHJ models misjudge slot filling or

Table 6. A positive example in CMRS corpus. The original utterance is ”30 号”, representing the 30th of a certain month.

input	30	号	
	Semantic tags		Intent label
gold	B-time	I-time	inf
+H models	B-pernum	I-time	inf
+C models	B-time	I-time	other
+C+H (CHJ) models	B-time	I-time	inf

Table 7. Some negative results transcripts of CHJ models. ”budget” and ”pernum” are different slot names, representing the budget-relevant and person-number-relevant slots respectively.

input	100		
	Semantic tags		Intent label
gold	B-pernum		inf
+C+H (CHJ) models	B-budget		inf
input	3000	people	
	Semantic tags		Intent label
gold	B-pernum	I-pernum	inf
+C+H (CHJ) models	B-budget	I-pernum	inf

intent classification. In the first example, the input utterance only comprises one number. In fact, the user was providing the attendance (pernum). Without the information of history utterances, it seems unlikely to label correctly. In the second example, ”B-budget” is followed by ”I-pernum”, which is illegal. CHJ models misjudge slot filling for lack of tag dependency. These defects provide a direction for our future work.

4 Conclusion

We have presented a cluster of CHJ models to jointly optimize slot filling and intent classification in NLU. The models are able to capture both contextual and hierarchical information in one joint structure. The combination of both kinds of information has been proved effective by comparison to other recent work. Finding the balance of two task losses is a great key to achieve best overall performances. We believe that CHJ models provide a novel hint for jointly learning subtasks of NLU.

There are several problems waiting for future work. For now, the lower supervision cannot affect the upper LSTM. In future work, we plan to figure out a more reasonable way for joint models that two losses could transmit supervision information equitably. Besides, we also want to incorporate tag dependency relations and history utterance information in our future work.

Acknowledgments. This paper is supported by 111 Project (No.B08004), NSFC (No.61273365), Beijing Advanced Innovation Center for Imaging Technology, Engineering Research Center of Information Networks of MOE, and ZTE.

References

1. Tur G, De Mori R. Spoken language understanding: Systems for extracting semantic information from speech[M]. John Wiley & Sons, 2011.
2. De Mori R, Bechet F, Hakkani-Tur D, et al. Spoken language understanding[J]. IEEE Signal Processing Magazine, 2008, 25(3).

3. Haffner P, Tur G, Wright J H. Optimizing SVMs for complex call classification[C]//Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on. IEEE, 2003, 1: I-I.
4. Yao K, Peng B, Zhang Y, et al. Spoken language understanding using long short-term memory neural networks[C]//Spoken Language Technology Workshop (SLT), 2014 IEEE. IEEE, 2014: 189-194.
5. Shi Y, Yao K, Chen H, et al. Contextual spoken language understanding using recurrent neural networks[C]//Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on. IEEE, 2015: 5271-5275.
6. Hakkani-Tür D, Tur G, Celikyilmaz A, et al. Multi-Domain Joint Semantic Frame Parsing using Bi-directional RNN-LSTM[C]// The, Meeting of the International Speech Communication Association. 2016.
7. Zhou Q, Wen L, Wang X, et al. A Hierarchical LSTM Model for Joint Tasks[C]//China National Conference on Chinese Computational Linguistics. Springer International Publishing, 2016: 324-335.
8. Yao K, Zweig G, Hwang M Y, et al. Recurrent neural networks for language understanding[C]//Interspeech. 2013: 2524-2528.
9. Mesnil G, Dauphin Y, Yao K, et al. Using recurrent neural networks for slot filling in spoken language understanding[J]. IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP), 2015, 23(3): 530-539.
10. Søgaard A, Goldberg Y. Deep multi-task learning with low level tasks supervised at lower layers[C]//Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, 2016, 2: 231-235.
11. Zhang X, Wang H. A Joint Model of Intent Determination and Slot Filling for Spoken Language Understanding[C]. IJCAI, 2016.
12. Liu B, Lane I. Attention-Based Recurrent Neural Network Models for Joint Intent Detection and Slot Filling[J]. arXiv preprint arXiv:1609.01454, 2016.
13. Hochreiter S, Schmidhuber J. Long short-term memory[J]. Neural computation, 1997, 9(8): 1735-1780.
14. Graves A. Supervised sequence labelling[M]//Supervised Sequence Labelling with Recurrent Neural Networks. Springer Berlin Heidelberg, 2012: 5-13.
15. Williams R J, Zipser D. Gradient-based learning algorithms for recurrent networks and their computational complexity[J]. Backpropagation: Theory, architectures, and applications, 1995, 1: 433-486.
16. Werbos P J. Backpropagation through time: what it does and how to do it[J]. Proceedings of the IEEE, 1990, 78(10): 1550-1560.
17. Graves A, Schmidhuber J. Framewise phoneme classification with bidirectional LSTM networks[C]//Neural Networks, 2005. IJCNN'05. Proceedings. 2005 IEEE International Joint Conference on. IEEE, 2005, 4: 2047-2052.
18. Ramshaw L A, Marcus M P. Text chunking using transformation-based learning[M]//Natural language processing using very large corpora. Springer Netherlands, 1999: 157-176.
19. Williams J, Raux A, Ramachandran D, et al. The dialog state tracking challenge[C]//Proceedings of the SIGDIAL 2013 Conference. 2013: 404-413.
20. Kim S, D' Haro L F, Banchs R E, et al. The fourth dialog state tracking challenge[M]//Dialogues with Social Robots. Springer Singapore, 2017: 435-449.
21. Duchi J, Hazan E, Singer Y. Adaptive subgradient methods for online learning and stochastic optimization[J]. Journal of Machine Learning Research, 2011, 12(Jul): 2121-2159.
22. Cotter A, Shamir O, Srebro N, et al. Better mini-batch algorithms via accelerated gradient methods[C]//Advances in neural information processing systems. 2011: 1647-1655.