Model	Features	ATIS		CQUD	
		Intent	Slot	Intent	Slot
SVM [Raymond and Riccardi, 2007]	W	—	89.76	—	81.32
CRF [Mesnil *et al.*, 2015]	W	—	92.94	—	83.40
CRF [Mesnil *et al.*, 2015]	W+N	—	95.16	—	—
RNN [Mesnil *et al.*, 2015]	W	—	95.06	—	85.63
RNN [Mesnil *et al.*, 2015]	W+N	—	96.24	—	—
R-CRF [Yao *et al.*, 2014]	W	—	—	—	**85.88**
R-CRF [Yao *et al.*, 2014]	W+N	—	**96.46**	—	—
Boosting [Tur *et al.*, 2010]	W	95.50	—	93.54	—
Sentence simplification [Tur *et al.*, 2011]	W+S	**96.98**	95.00	**94.46**	—
RecNN [Guo *et al.*, 2014]	W+S	95.40	93.22	—	—
RecNN+Viterbi [Guo *et al.*, 2014]	W+S	95.40	93.96	—	—
Our model	W	98.10	95.49	**96.05**	**87.12**
Our model	W+N	**98.32**	**96.89**	—	—

Table 2: Comparison with previous approaches

than words. Nevertheless, the improvements are consistent in CQUD. Our model outperforms the state-of-the-art methods by 1.59% for ID and 1.24% for SF.

4.6 Joint Model vs Separate Models

First, we give the definitions of joint model and separate models. The joint model is our proposed model in Figure 1. The separate model is similar to the joint model except that there is only one task. For ID, there is only the shared layers and ID specific layers without SF specific layers. It is in the same way for SF. We also implement a pipeline method. First a RNN is trained for ID, and then the predicted intent is used as addition feature to train another RNN for SF.

The speed advantage of the joint model is self-evident, because only one model is needed to train and test. The shared part of the model is only calculated one time for the two tasks. For quantitative analysis, we ran programs of the joint model and separate models using same parameter settings and hardware. On ATIS dataset, the time for training one epoch using joint model is 124 seconds, while the sum of time using separate models is 212 seconds.

Next we compare the performance of the joint model and separate models. In these experiments, only lexical features are used. Here a new concept, joint and with one task oriented, is introduced. In the joint model, we can pay different attentions to the two tasks. This is achieved by adjusting the weight factor α in Equation 20 and using score of one task as target to select hyper-parameters. Larger α means that more attention is paid to SF. The results are listed in Table 3.

The joint model outperforms separate models for two tasks, showing that the joint training is effective. The correlations of two tasks are learned by our joint model and contribute to the two tasks. Because of the two-way information sharing and supervision, our joint model outperforms the one-way pipeline method. Note that if we set one task as oriented in the joint model, higher performance can be acquired for it comparing to treating two tasks equally. This brings flexibility to have tendency to one task if high score is required for it or even only one task is needed in a real application.

Model	ATIS		CQUD	
	Intent	Slot	Intent	Slot
ID only	97.53	—	95.34	—
SF only	—	95.14	—	85.78
Pipeline	97.53	95.41	95.34	86.96
Joint (equal)	98.10	95.49	96.05	87.12
Joint (ID oriented)	98.10	95.49	**96.35**	86.63
Joint (SF oriented)	97.87	**95.61**	95.93	**87.23**

Table 3: Comparison of joint model and separate model

In ATIS, α is set to 1.6 for equal model, 1.6 for ID oriented and 1.8 for SF oriented. In CQUD, α is set to 1.5, 2.0 and 1.8 respectively for three models. Intuitively, the performance of one task gets better with higher weight on it. It is not always true in our experiments, which may be because too large weight for one task leads to too quick convergence such that parameters are not well tuned for that task.

5 Conclusion and Future Work

In this paper we have introduced recurrent neural networks for joint intent determination and slot filling, which are two major tasks in spoken language understanding. Bidirectional GRUs are used to learn the sequence representations shared by two tasks. A global representation is acquired by a max-pooling of the shared representations to predict the label of intent. The labels of slots are predicted by the shared representations and are further inferred at sequence level. Through a united loss function and shared representations, the correlations of the two tasks are learned so as to promote each other. We conducted experiments on two datasets. The joint model demonstrates advantages over separate models and outperforms the state-of-the-art approaches on both tasks.

In future works, we plan to improve our model by using syntactic information. Furthermore, our CQUD dataset is still small-scale for the application of deep learning methods. We would like increase the scale of our dataset, which can be useful for SLU and QA research.