Context Sensitive Spoken Language Understanding Using Role Dependent LSTM Layers

Chiori Hori **Takaaki Hori** **Shinji Watanabe** **John R. Hershey**

Mitsubishi Electric Research Laboratories
Cambridge MA, USA
{chori, thori, watanabe, hershey}@merl.com

Abstract

Neural network models have become a recent focus of investigation in spoken language understanding (SLU). To understand speaker intentions accurately in a dialog, it is important to consider the sentence in the context of the surrounding sequence of dialog turns. In this study, we use long short-term memory (LSTM) recurrent neural networks (RNNs) to train a context sensitive model to predict sequences of dialog concepts from the spoken word sequences. In this model, words of each utterance are input one at a time, and concept tags are output at the end of each utterance. The model is trained from human-to-human dialog data annotated with concept tags representing client and agent intentions for a hotel reservation task. The LSTM layers jointly represent both the context within each utterance, and the context within the dialog. The different roles of client and agent are modeled by switching between role-dependent layers. To evaluate the performance of our models, we compared label accuracies using Logistic Regression (LR) and LSTMs. The results show 70.8% for LR, 72.4% for LR w/ word2vec, 78.8% for context sensitive LSTMs, and 84.0% for role dependent LSTMs. We confirmed significant improvement by using context sensitive role dependent LSTMs.

1 Introduction

Spoken language understanding (SLU) methods are used in dialog systems to estimate the intention of user utterances obtained from an automatic speech recognition (ASR) system [1, 2]. Conventional intention estimation approaches are either based on phrase matching, or traditional classification methods such as boosting, support bector machines (SVM), and logistic regression(LR), using bag of word (BoW) features as inputs.

Recently, recurrent neural networks (RNNs) have been actively investigated to utterance classification to consider history of a word sequence in each utterance [3, 4, 5]. Furthermore, long short-term memory (LSTM) RNNs were applied to spoken language understanding[5]. However, these models were only used for word sequence context within an utterance without considering the broader context of the sequence of utterances. One might expect that the speaker intentions of each utterance can be more accurately inferred, especially in dialogs, if the context of the utterance within the dialog is also taken into account. This hypothesis appears to be borne out in previous work: context sensitive understanding using phrase matching, weighted finite state transducer-based dialog management (WFSTDM) was previously proposed [6]. More recently, conventional RNNs considering contextual information were applied to domain and intention classification [7], intention classification, and goal estimation [8] and system response generation [9].

LSTMs are a form of RNN designed to improve learning of long-range context, and have been shown to be effective for difficult problems [10]. In this study, we apply LSTMs to capture long-

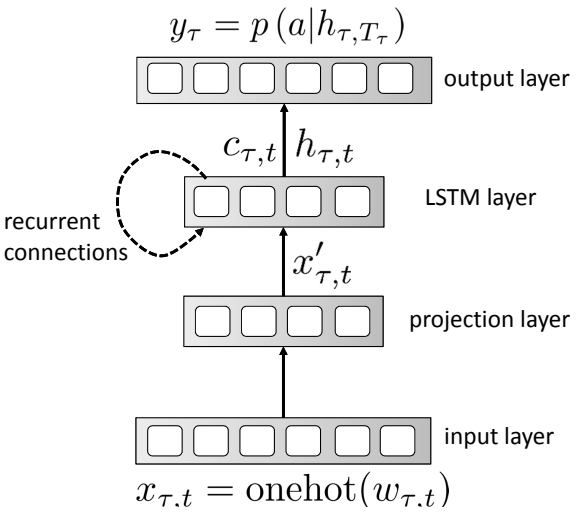

Figure 1: Recurrent Neural Network.

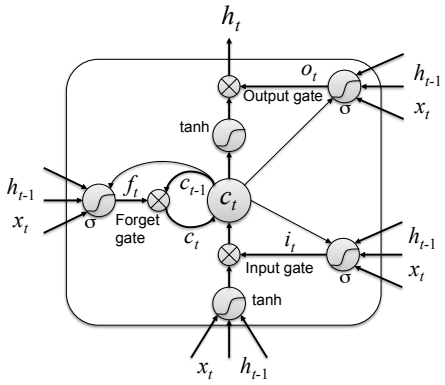

Figure 2: LSTM cell

term characteristics over an entire dialog. Each word is input sequentially into a LSTM and concept tags are output at the end of each utterance. To propagate contextual information through a dialog, the activation vector of the LSTM for an utterance serves as input to the LSTM for the next utterance. In this study, the LSTMs were trained from a human-to-human dialog corpus annotated with concept tags which represent client and agent intentions for hotel reservation. The expressions of utterances in the dialog corpus are characterized by each role of agent and client. In order to precisely model the role dependent expressions, we introduce two parallel LSTM layers representing client and agent expressions.

2 Context-sensitive SLU using LSTMs

The model we use for context-sensitive spoken language understanding is a recurrent neural network depicted in Fig. 1. The network has an input layer that takes each input word, a projection layer that reduces the word vector in a low-dimensional space, a hidden layer with recurrent connections that keeps context information, and an output layer that estimates posterior probabilities of output labels. In the hidden layer, we use a set of LSTM cells instead of regular network units. In theory, an LSTM cell can remember a value for an arbitrary length of time due to a system of gating. The LSTM cell contains input, forget, and output gates which determine when the input is significant enough to remember, when it should continue to remember or forget the value, and when it should contribute to the output value. An example of an LSTM cell is depicted in Fig. 2.

Suppose, given a sequence of M utterances, $u_1, \ldots, u_\tau, \ldots, u_M$, each utterance consists of word sequence $w_{\tau,1}, \ldots, w_{\tau,t}, \ldots, w_{\tau,T_\tau}$ and its concept tag (or dialog act) a_τ. The input vector $x_{\tau,t}$ is prepared as

$$x_{\tau,t} = \text{OneHot}(w_{\tau,t}), \tag{1}$$

where word $w_{\tau,t}$ in vocabulary $\mathcal{V}$ is converted by 1-of-N coding using function OneHot(w), i.e. $x_{\tau,t} \in \{0,1\}^{|\mathcal{V}|}$.

The input vector is projected to the D dimensional vector

$$x'_{\tau,t} = W_{pr} x_{\tau,t} + b_{pr} \tag{2}$$

and fed to the recurrent hidden layer, where W_{pr} and b_{pr} are the projection matrix and the bias vector.

At the hidden layer, activation vector $h_{\tau,t}$ is computed using the LSTM cells according to the way of [11][12], i.e.

$$\begin{aligned}
i_{\tau,t} &= \sigma(W_{xi} x'_{\tau,t} + W_{hi} h_{\tau,t-1} + W_{ci} c_{\tau,t-1} + b_i) && (3)\\
f_{\tau,t} &= \sigma(W_{xf} x'_{\tau,t} + W_{hf} h_{\tau,t-1} + W_{cf} c_{\tau,t-1} + b_f) && (4)\\
c_{\tau,t} &= f_{\tau,t} c_{\tau,t-1} + i_{\tau,t} \tanh(W_{xc} x'_{\tau,t} + W_{hc} h_{\tau,t-1} + b_c) && (5)\\
o_{\tau,t} &= \sigma(W_{xo} x'_{\tau,t} + W_{ho} h_{\tau,t-1} + W_{co} c_{\tau,t} + b_o) && (6)\\
h_{\tau,t} &= o_{\tau,t} \tanh(c_{\tau,t}), && (7)
\end{aligned}$$

where $\sigma()$ is the element-wise sigmoid function, and $i_{\tau,t}$, $f_{\tau,t}$, $o_{\tau,t}$ and $c_{\tau,t}$ are the input gate, forget gate, output gate, and cell activation vectors for the t-th input word in the τ-th utterance, respectively. The weight matrices W_{zz} and the bias vectors b_z are identified by the subscript $z \in \{x, h, i, f, o, c\}$. For example, W_{hi} is the hidden-input gate matrix and W_{xo} is the input-output gate matrix.

The output vector is computed at the end of each utterance as

$$y_\tau = \text{softmax}(W_{HO} h_{\tau,T_\tau} + b_O), \tag{8}$$

where W_{HO} and b_O are the transformation matrix and the bias vector to classify the input vector into different categories according to the hidden vector. softmax$()$ is an element-wise softmax function that converts the classification result into label probabilities, i.e. $y_\tau \in [0,1]^{|\mathcal{L}|}$ for label set $\mathcal{L}$.

$$\hat{a}_\tau = \underset{a \in \mathcal{L}}{\text{argmax}}\, y_\tau[a], \tag{9}$$

where $y_\tau[a]$ indicates the component of y_τ for label a, which corresponds to label probability $P(a|h_{\tau,T_\tau})$.

To inherit the context information from the previous utterances, the hidden and cell activation vectors at the beginning of each utterance are

$$\begin{aligned}
h_{\tau,0} &= h_{\tau-1,T_{\tau-1}} && (10)\\
c_{\tau,0} &= c_{\tau-1,T_{\tau-1}}, && (11)
\end{aligned}$$

where $\tau > 1$ and $h_{1,0} = c_{1,0} = 0$

Figure 3 illustrates a propagation process of our context-sensitive SLU. Words are sequentially input to the LSTM and an output label corresponding to the utterance concept at the end of the utterance, where symbol "EOS" stands for a sentence end. This model is similar to the LSTM used in [5] for SLU, but it considers the entire context from the beginning of the dialog, while the model in [5] considers each utterance independently. Accordingly, the label probabilities can be inferred using not only the sentence-level intentions but also the dialog-level context.

3 Role-dependent LSTM layers

In this study, the LSTMs were trained from a human-to-human dialog corpus annotated with concept tags which represent client and agent intentions for hotel reservation. The expressions are characterized by each role of agent and client. In order to precisely model the role dependent expressions,

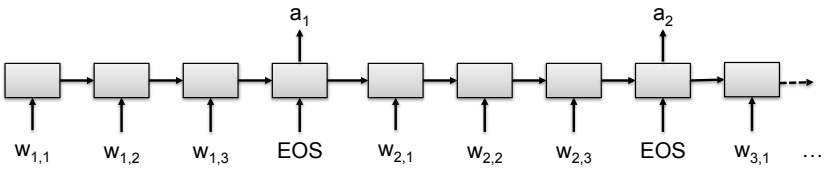

Figure 3: Propagation through time in context-sensitive SLU

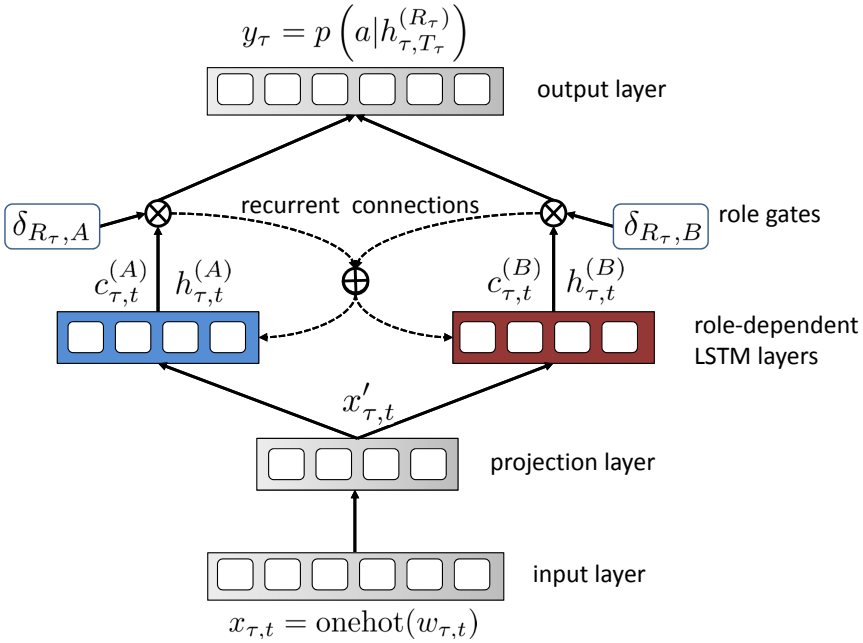

Figure 4: LSTM with role-dependent layers. The blue layer (A) corresponds to client utterance states and the red layer (B) corresponds to agent utterance states. Role gates control which role is active.

two parallel LSTM layers representing client and agent expressions are incorporated in the model as shown in Fig. 4.

The two LSTM layers have different parameters depending on the speaker roles. The input vector is thus processed differently by the left layer for the client's utterances, and by the right layer for the agent's utterances. The active role for a given utterance is controlled by a role variable $\mathcal{R}$, which is used to gate the output of each LSTM layer. The gated output then passes both to the recurrent LSTM inputs and to the output layer. The recurrent LSTM inputs thus receive the output from the role-dependent layer active at the previous frame, allowing for transitions between roles. Error signals in the training phase are also back-propagated through the corresponding layers. Here, we assume that the role of each speaker does not change during a dialog and it is known which speaker uttered which utterance. However, the model structure leaves open the possibility of dynamically inferring the roles. Accordingly, we can compute the activation at the output layer as

$$y_\tau = \text{softmax}\left(\delta_{\mathcal{R},R_\tau}(W_{HO}h^{(\mathcal{R})}_{\tau,T_\tau} + b_O)\right), \tag{12}$$

where $h^{(\mathcal{R})}_{\tau,T_\tau}$ is the hidden activation vector given by the LSTM layer of role $\mathcal{R}$, and $\delta_{\mathcal{R},R_\tau}$ is Kronecker's delta, i.e. if R_τ the role of the τ-th utterance equals role $\mathcal{R}$, it takes 1, otherwise takes 0. Furthermore, at the beginning of each utterance, the hidden and cell activation vectors of the role-dependent layer are given as

$$h^{(R_\tau)}_{\tau,0} = h^{(R_{\tau-1})}_{\tau-1,T_{\tau-1}} \tag{13}$$

$$c^{(R_\tau)}_{\tau,0} = c^{(R_{\tau-1})}_{\tau-1,T_{\tau-1}}. \tag{14}$$

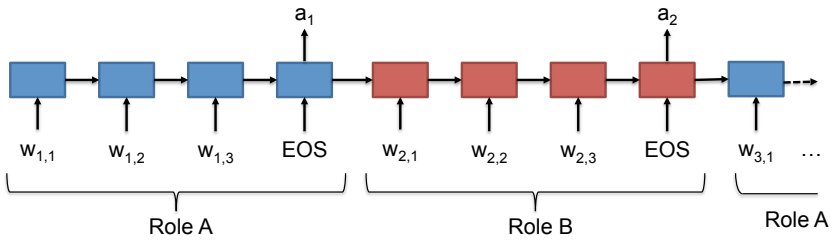

Figure 5: Propagation through time in context-sensitive role-dependent SLU. The blue boxes correspond to client utterance states and the red boxes correspond to agent utterance states.

Table 1: An Example of Hotel Reservation Dialog

Speaker	Utterance	Concept tags
Agent	hello ,	greeting
Agent	new york city hotel ,	introduce-self
Agent	may i help you ?	offer+help
Cleint	i would like to make a reservation for a room .	request-action+reservation+hotel
Agent	very good .	acknowledge
Agent	may i have the spelling of your name , please ?	request-information+name
Client	it is m i k e , s m i t h .	give-information+name
Agent	uh when would you like to stay ?	request-information+temporal
Client	from august tenth to august sixteenth , seven days .	give-information+temporal
Agent	i see ,	acknowledge
Agent	you would be arriving on the tenth ? is that right ?	verify-give-information+temporal
Client	that is right .	affirm
Agent	great .	acknowledge
Agent	and , what sort of room would you like ?	request-information+room
Client	well , it is just for myself ,	give-information+party
Client	so a single room would be good .	give-information+room
Agent	okay .	acknowledge
Agent	a single ,	verify-give-information+room
Agent	starting on the tenth and	verify-give-information+temporal
Agent	you would be checking out on the sixteenth ? is that right ?	verify-give-information+temporal
Client	yes ,	affirm
Client	and i like the second or third floor , if possible .	give-information+room
Agent	i see .	acknowledge

Figure 5 shows the temporal process of the role-dependent SLU. For each utterance in a given role, only the LSTM layer for that role is active, and the hidden activation and the cell memory are propagated over dialog turns. In the figure, the blue boxes correspond to client utterance states and the red boxes correspond to agent utterance states. With this architecture, the both layers can be trained considering a long context of each dialog, and the model can predict role-dependent concept labels more accurately.

Table 2: Label Accuracies

	Dialog Act (DA)		Dialog Act + Slot type (DA+ST)	
	Dev. set	Test set	Dev. set	Test set
Logistic Regression	69.8	70.8	61.4	61.6
Logistic Regression w/ word2vec	71.1	72.4	62.1	62.3
Utterance-based LSTM	73.3	69.8	59.5	56.2
Context-Sensitive LSTM	**81.3**	**78.8**	**64.7**	**64.5**
+Role dependent layers	**84.6**	**84.0**	**69.4**	**70.3**

4 Experiments

4.1 Dialog Data

We trained models using a human-to-human dialog data annotated with concept tags representing client and agent intentions for hotel reservation. Table 1 shows samples of the utterances and tags used for the task. In the experiments, Japanese utterances were used. 131 dialogs were split into 97 dialogs (5213 utterances) for training, 17 dialogs (1006 utterances) for development sets and 17 dialogs (1134 utterances)for test sets. The vocabulary size of the training data is 1550. The concept tags are based on Interchange Format (IF) which is an Interlingual for speech translation systems. The original tags indicates a combination of dialog acts, slot types and slot values. To model dialog discourses, two different layers of tags are used. One is a combination of dialog acts and slot types such as "request-information+room". the total number of the tags is 419 consisting of 186 client and 233 agent tags. The other one is dialog acts layer only such as "request-information" in which consists of 65 tags including 29 client and 36 agent tags.

4.2 Classifiers

To evaluate efficiency of our proposed method, we compared label accuracies using Logistic Regression (LR) and LSTMs, including the context sensitive LSTMs with and without the role dependent LSTM layer. In the baseline LR system, each sentence is represented as a bag-of-Words feature vector. We tested the performance of word2vec features [13] by concatenating the bag-of-words and word embedding features [8]. The 200-dimensional word2vec features were used for the "dialog act" (DA) LR system and 500-dimensional word2vec features were used for the "dialog acts and slot type" (DA+ST) LR system, where the dimensions were selected using the dev. set. We used a 1.8G Japanese web text corpus to train word2vec in unsupervised fashion. The context sensitive LSTMs and speaker role based LSTM layers were trained on the platform of Chainer [14].

4.3 Evaluation Results

The experimental results are shown in Table 2. By using the word2vec features, the performance was slightly improved from the baseline system by 1.3% (dev.) and 1.6% (test) for DA and 0.7 % (dev. and test) for DA+ST. The performance further improves to 10.2%(dev.) and 6.4%(test) for DA and 2.6%(dev.) and 1.2%(test) for DA+ST by the context sensitive LSTMs. This result confirms the importance of modeling dialog context in SLU. In addition, the speaker role based LSTM layers significantly improve the performance from the context sensitive LSTMs only to 3.3%(dev.) and 5.2%(test) for DA and 4.7%(dev.) and 5.8%(test) for DA+ST. These results indicate that roll-dependent LSTM layers which characterize expressions of utterances varied among each role contribute to intention classifier.

5 Conclusion

We proposed an efficient context sensitive SLU approache using role-based LSTM layers. In order to capture long-term characteristics over an entire dialog, we implemented LSTMs representing intention using consequent word sequences of each concept tag. We evaluated the performance of importing contextual information of an entire dialog for SLU and the effectiveness of the speaker role based LSTM layers. The context sensitive LSTMs with roll-dependent layers out-performs utterance-based approaches and improves the SLU baseline by 11.6% and 8.0% (absolute) for the layer of DA and DA+ST, respectively. In this study, LSTMs are trained from the features of word sequences only to predict the concept tags. Future works will test LSTMs trained using the feature of the concept tags explicitly to improve label accuracies.

Acknowledgements

The authors thank Prof. Alex Waibel of Karlsruhe Institute of Technology and Carnegie Mellon University to let us test our proposed method using the hotel reservation dialogs with Interchange Format.

References

[1] Dan Jurafsky and James H Martin, *Speech & Language Processing*, Pearson Education, 2000.

[2] Renato De Mori, "Spoken language understanding: a survey.," in *ASRU*, 2007, pp. 365–376.

[3] Kaisheng Yao, Geoffrey Zweig, Mei-Yuh Hwang, Yangyang Shi, and Dong Yu, "Recurrent neural networks for language understanding.," in *INTERSPEECH*, 2013, pp. 2524–2528.

[4] Kaisheng Yao, Baolin Peng, Geoffrey Zweig, Dong Yu, Xiaolong Li, and Feng Gao, "Recurrent conditional random field for language understanding," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 4077–4081.

[5] Kaisheng Yao, Baolin Peng, Yu Zhang, Dong Yu, Geoffrey Zweig, and Yangyang Shi, "Spoken language understanding using long short-term memory neural networks," in *Spoken Language Technology Workshop (SLT)*, December 2014.

[6] Chiori Hori, Kiyonori Ohtake, Teruhisa Misu, Hideki Kashioka, and Satoshi Nakamura, "Statistical dialog management applied to WFST-based dialog systems," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2009.

[7] Yangyang Shi, Kaisheng Yao, Hu Chen, Yi-Cheng Pan, Mei-Yuh Hwang, and Baolin Peng, "Contextual spoken language understanding using recurrent neural networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

[8] Shinji Watanabe Yi Luan and Bret Harsham, "Efficient learning for spoken language understanding tasks with word embedding based pre-training," in *Proc. Interspeech2015*, 2015.

[9] Ryan Lowe, Nissan Pow, Iulian Serban, and Joelle Pineau, "The ubuntu dialogue corpus A large dataset for research in unstructured multi-turn dialogue systems," in *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*.

[10] Ilya Sutskever, Oriol Vinyals, and Quoc VV Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[11] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[12] A. Graves, A.-R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, May 2013, pp. 6645–6649.

[13] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[14] Preferred Networks, "Chainer," in *"http://chainer.org/"*.