ASTES

Advances in Science, Technology and Engineering Systems Journal Vol. 2, No. 3, 1741-1747 (2017)

www.astesj.com

Special Issue on Recent Advances in Engineering Systems

ASTESJ

ISSN: 2415-6698

Retrieving Dialogue History in Deep Neural Networks for Spoken Language Understanding

Myoung-Wan Koo*, [1], Guanghao Xu[1], Hyunjung Lee[2], Jungyun Seo[1]

[1]*Department of Computer Science and Engineering, Sogang University, 04107, Republic of Korea*

[2]*Institut für Linguistik, Universität Leipzig, 04107, Germany*

ARTICLE INFO

Article history:
Received: 30 May, 2017
Accepted: 13 August, 2017
Online: 15 September, 2017

Keywords:
Spoken Language Understanding
Convolutional Neural Network
Recurrent Neural Network

ABSTRACT

In this paper, we propose a revised version of the semantic decoder for multi-label classification task in the spoken language understanding (SLU) pilot task of the Dialog State Tracking Challenge 5 (DSTC5). Our model concatenates two deep neural networks - a Convolutional Neural Network (CNN) and a Recurrent Neural Networks (RNN) - for detecting semantic meaning of incoming utterance with the assistance of algorithm adaptation method. In order to evaluate the robustness of our proposed models, comparative experiments on the DSTC5 dialogue datasets are conducted. Experimental results show that the proposed models outperform most of the submitted models in the DSTC5 in terms of F1-score. Without any manually designed features or delexicalization, our model has proven its efficiency of tackling the multi-label SLU task, using only publicly available pre-trained word vectors. Our model is capable of retrieving the dialogue history, and thereby it could build the concise concept structure by employing the pragmatic intention as well as semantic meaning of utterances. The architecture of our semantic decoder has a potential to be applicable to other variety of human-to-human dialogues to achieve SLU.

1. Introduction

The spoken language understanding (SLU) has been one of the fundamental components of an end-to-end dialogue system. The Dialog State Tracking Challenge 5 (DSTC5) released a pilot SLU task, which requires to extract semantic meaning of users' utterances in task-oriented dialogues and to fill the slots with speech acts. Unlike the previous challenges (DSTC 2&3) where the human-to-system dialogues were targeted, the corpus of DSTC 5 has been no more than challengeable due to the following points: human-to-human dialogues, cross-linguistic data and multi-label classification task. As its corpus is built by collecting human-to-human conversation in a natural setting, more than one speech act can be annotated to a single utterance.

Xu et al. propose a Convolutional Neural Network (CNN) model with a threshold predictor to tackle a multi-label speech act classification task on DSTC5 corpus [1]. With the assistance of algorithm adaptation method, the model they propose is adapted for the multi-label classification task without any manually designed features. Although the model, however, has advantage of handling the multi-label classification task, there still remain rooms for improvement to achieve state-of-art SLU.

The aim of this study is to improve the model proposed by Xu et al. by building a more robust and concise semantic classifier. In this revised model two deep neural networks, Convolutional Neural Network and Recurrent Neural Network, are conjoined to conduct a multi-label speech act classification task in an improved fashion. Our newly revised model shows a synergy effect of retrieving dialog context as well as exploiting a current utterance. In addition, a threshold learning mechanism is engaged to enable our proposed model to produce an output of a set of multiple labels called speech acts.

The rest of this paper is organized as follows. Section 2 gives a brief review of DSTC5 dataset and some related works in the SLU pilot task. In Section 3, we introduce the architecture of our proposed model and a threshold predictor. The Section 4 gives a detailed description of the DSTC5 dataset and describes how we set up the experiments for training data and evaluation process. In Section 5, we provide our experimental results to optimize the

*Corresponding Author: Myoung-Wan Koo, Department of Computer Science and Engineering, Sogang University, Republic of Korea, mwkoo@sogang.ac.kr

Table 1. An example of test utterances annotated with speech act information.

Speaker	Chinese Utterances and their Translated English Sentences.	Speech Act Category (Attribute)
Guide	um, sentosa the universal studios in the matter. You see it, the whole family. 嗯，圣淘沙里面的环球影城啦，你看啦，一家大小。	FOL (ACK) FOL (INFO)
Tourist	there are still in the place where I can recommend? 还有地方可以介绍的吗？	QST (RECOMMEND) QST (WHERE)
Guide	yes, we have, um, the zoo. the daytime the zoo. 嗯，我们有一个动物园，那个日间动物园。	FOL (RECOMMEND) FOL (WHERE)
Tourist	how big is the singapore? 新加坡有多大？	QST (INFO)

performance of our CNN-RNN classifier on the DSTC5 SLU task. The Section 6 discusses the differences between the previous and current models and the strength of concatenating two deep neural networks in SLU task. The Section 7 concludes this paper[1].

2. Background

2.1. The DSTC5 Dataset

The DSTC5 provides the TourSG corpus, which consists of dialogue sessions collected from Skype calls between tour guides and tourists focusing on offering touristic information of Singapore[2]. For the SLU task, the system is given the utterances from both the tourist and the guide as its input, and the system subsequently tags the utterances spoken by both the speakers with appropriate *speech acts categories* and *attributes*.

Each sub-utterance belongs to one of the four basic *speech act categories* that denote general information of current dialogue flow. More specific speech act information can be annotated by the combination with the *speech act attributes*. Therefore, a classifier is demanded for classifying a set of labels consisting of speech act categories and attributes tagged to a single utterance. Reference [3] gives complete list of speech act categories and attributes.

Table 1 shows Chinese test utterances and ones translated in English that annotated with their corresponding speech act categories and attributes.

2.2. Other models in pilot SLU task of the DSTC5

A simple baseline model for the SLU task is provided by the committee of the DSTC 5, which uses a binary relevance (BR) approach and trains a set of linear support vector machines (SVM) for multi-label speech act classification. The baseline model is built within traditional TF-IDF approach which mainly depends on keywords that appeared in the utterances per speaker. This baseline model, however, has a crucial deficiency in detecting semantic meanings of utterances appropriately, as it only superficially decodes meaning relying on words on the surface level.

Ushio et al., which takes the first place in this challenge, proposes a local co-activate multi-task learning model (LC-MTL) for capturing structured speech acts by using recurrent convolutional neural networks[4]. This model consists of a CNN, which represents incoming utterances as sentence vectors, and two LSTMs, which locally co-activates neurons in hidden layer between speech act categories and attributes of the corresponding utterances.

Xu et al also proposes a CNN model with a threshold predictor, which enables to predict more than one speech act annotated to an utterance. It has been proven that CNN is capable to detect the core semantic meaning without any knowledge on the syntactic structures of a language or any manually designed features. This model, however, has faced the ambiguity issue, as the output node of CNN only depend on each current utterance to produce proper scores for each set of labels. In Section 3 we will introduce the revised version of the semantic decoder that has a power to disambiguate the meaning of utterances.

2.3. Related Works

So far there have been many researches grounded on CNN architecture with regard to the language processing tasks. The CNN proposed by Kim achieves good performance across several datasets despite of its simple architecture which consists of a convolutional layer, a max-pooling layer and a softmax classifier. Wang et al. proposed another architecture of CNN with an additional semantic layer, which exploits the contextual information from short texts[6]. Another CNN model with an unsupervised “region embedding”, proposed by Johnson, works well for long text classification task like movie reviews[7]. Zhang et al. explored the effects of hyperparameters in one-layer CNN architectures, and reported their impact on performance over multiple runs[8].

In addition, recent advances in deep learning with RNN including LSTM have also achieved impressive improvements on various natural language processing tasks such as language modeling, sequence tagging and machine translation. Language model based on RNN(RNN-LM) was proposed by Mikolov et al., which significantly outperformed the previous back-off models, even in case when RNN-LM was trained on much less data than back-off models were [9]. Another variant of RNN-LM based on LSTM proposed by Sundermeyer, showed additional improvements of 8% relative in perplexity over the RNN-LM[10]. Huang et al. proposed a sequence tagging model using a bidirectional LSTM, which was capable of using effectively both previous and oncoming input features. Their model have achieved state of the art accuracy on various sequence tagging tasks such as POS tagging, chunking and named entity recognition(NER)[11]. Sutskever et al. proposed a sequence to sequence learning model

[1] The source code of the proposed system is available on https://github.com/hkhpub/dstc5-slu

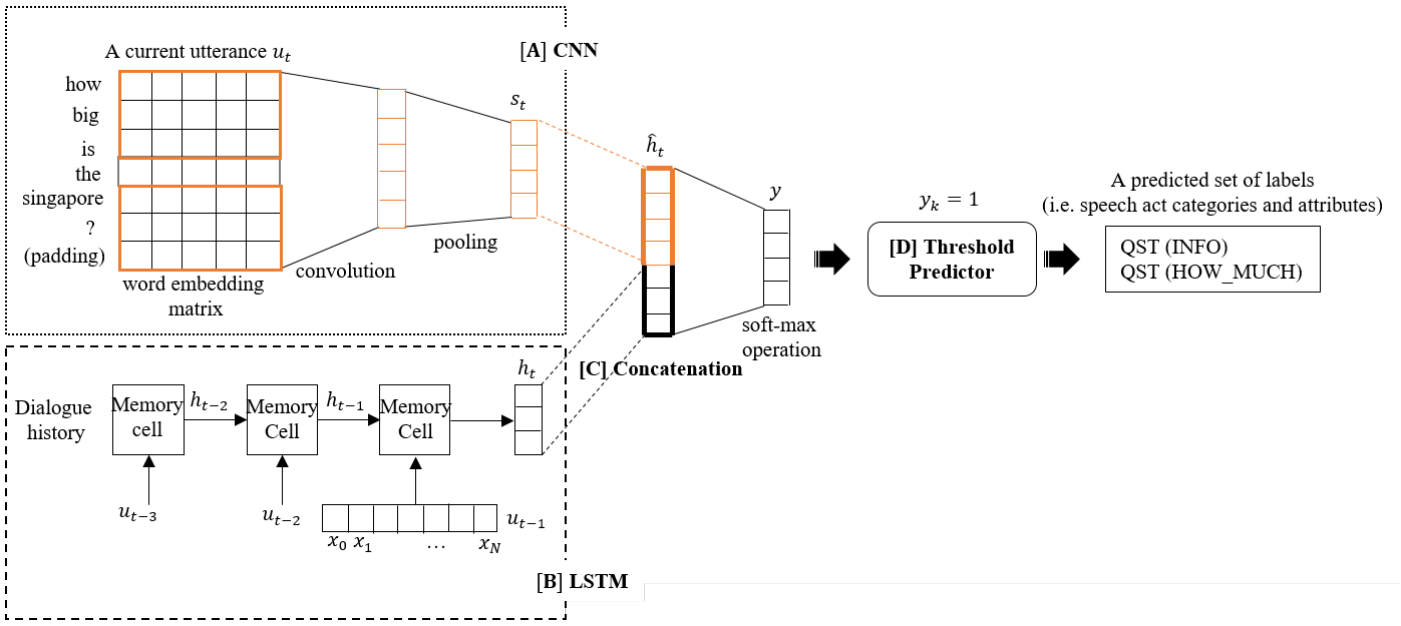

Figure 1. The architecture of our proposed model.

grounded on LSTM. They achieved a higher BLEU score on a translation task from English to French on the WMT-14 dataset, compared to a phrase-based machine translation system[12].

Several researches have been recently conducted by combining CNN and RNN models. Vinyals et al. proposes an conjoined model for image captioning, which encodes image features using deep convolutional networks and automatically generates captions with recurrent networks[13]. Kim proposes a recurrent convolutional network model, in which the penultimate layer of CNN is connected to the recurrent layers of the RNN model in order to track a topic of a dialogue segment in human-human conversations[14]. Unlike the previous CNN-RNN models, another jointed CNN and RNN model is proposed by Barahona et al., where the model is optimized with two distinctive inputs: a current user' utterance and dialog act-slot pairs of previous system's utterances [15]. In the task of decoding semantic meaning of spoken languages each input is utilized in sentence representation and context representation, respectively.

3. Model Architecture

In this section, we propose a jointed model which combines Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) . The overall architecture of our semantic decoder is depicted in Figure 1. Our model consists of three modules: (i) a CNN with multiple filters that encodes semantic meaning of current utterance, output nodes that produces scores for each label, (ii) a RNN with recurrent connections between neurons that store contextual information of dialogues and (iii) a multi-label threshold predictor that generates a reference point using the scores of the labels. The threshold is then used to for the system to decide whether each label is as relevant or irrelevant. Specifically, we newly adopt the RNN to improve the previous semantic decoder of Xu et al. with assistance of contextual information drawn from dialogues so that the disambiguity problem is raveled out.

3.1. CNN Architecture

Coming up with the architecture of CNN [5], which is specifically illustrated in the part [A] of Figure 1, given our CNN classifier is capable of optimizing its parameters with respect to multi-label cross entropy loss function. Formally, let $x_i \in \mathbb{R}^k$ be the k-dimensional word embedding vector corresponding to i-th word in a given utterance. An utterance u_t of length n are represented as a $n \times k$ matrix:

$$u_t = x_1 \oplus x_2 \oplus \ldots \oplus x_n \quad (1)$$

where t is the index of dialogue turn and $\oplus$ is the concatenation operator. A convolutional operation involves a filter $w \in \mathbb{R}^{h \times k}$, which is applied to a window of h adjacent words to produce a new feature. A feature c_i is generated by applying a hyperbolic tangent function f:

$$c_i = f(w \cdot x_{i:i+h-1} + b) \quad (2)$$

where $b \in \mathbb{R}$ is a bias term. The filter is applied to every possible window of words in the utterance to produce a feature map:

$$c = [c_1, c_2, \ldots, c_{n-h+1}] \quad (3)$$

A max-over-time pooling is then operated to take the maximum value $\hat{c} = \max\{c\}$ as a representative feature for the filter.

Following the same procedure as described above, multiple filters with varying window size h are integrally engaged into multiple adjacent features. These features are then concatenated into a fixed-length and 'top-level' feature vector s_t which automatically encodes most of representative features from a given utterance at dialogue turn t.

3.2. RNN Architecture

Since each utterance is dependent on the previous utterances in a conversation, referring to dialogue context from previous dialogue utterances is essential to wholly understand the meaning

Table 2. Statistics of dstc5 datasets.

Datasets (Speaker)	Language	*M*	*L*	*C*
Train (Tourist)	English	14,226	74/88	1.19
Test (Tourist)	Chinese	4,085	61/88	1.16
Train (Guide)	English	19,916	69/88	1.24
Test (Guide)	Chinese	8,555	71/88	1.21

M: Number of utterances.
L: Size of label set (size/total).
C: Average number of labels per utterance.

of a current utterance. To access to dialogue context, we employ a long short term memory (LSTM). It is much better for preserving information over long periods of time than vanilla RNN due to its ability to deal with vanishing and exploding gradients[16].

To track the dialogue context, two questions must be taken into consideration: (i) to what extent we should track previous utterances as dialogue history and (ii) in which form those utterances are fed into the LSTM as inputs. For length L of the dialogue history, we treat it as a free parameter and determine its value empirically. Once the length L is determined, the dialogue history is represented as $\{u_{t-L}, u_{t-L+1}, \dots, u_{t-1}\}$ where t is the index of dialogue turn.

As depicted in the part [B] of Figure 2, the structure of LSTM is divided into a memory cell c_l and three gates: a forget gate f_l, an input gate i_l and an output gate o_l. Three kinds of gates functions to decide which amount of information the memory cell should keep or forget at a time step l, where l denotes for each word in the utterances given a length L. The input x_l and the output h_l of LSTM are updated as follows:

$$i_l = \sigma(W^i \cdot x_l + U^i \cdot h_{l-1} + b^i) \quad (4)$$

$$f_l = \sigma(W^f \cdot x_l + U^f \cdot h_{l-1} + b^f) \quad (5)$$

$$o_l = \sigma(W^o \cdot x_l + U^o \cdot h_{l-1} + b^o) \quad (6)$$

$$g_l = tanh(W^g \cdot x_l + U^g \cdot h_{l-1} + b^g) \quad (7)$$

$$c_l = f_l \odot c_{l-1} + i_l \odot g_l \quad (8)$$

$$h_l = o_l \odot tanh\,(c_t) \quad (9)$$

where x_l is the input at the current time step, h_l is the hidden unit at time step l, b is a bias term, $\sigma(\cdot)$ is a logistic sigmoid function and $\odot$ denotes a point-wise multiplication operation. Each word of the utterances in the dialogue history is represented as word embedding vectors and fed into the LSTM sequentially as an input x_l. The last hidden unit h_l of LSTM is obtained which encodes the dialogue context information for current utterance u_t.

3.3. Combining CNN and LSTM

To utilize both the representative feature of current utterance and context information of previous utterances, we concatenate the hidden unit h_t of LSTM to the 'top-level' feature vector s_t modeled by the CNN, as illustrated in [C] of Figure 1. Then, the penultimate layer consists of the concatenated vector $\hat{h}_t = s_t \oplus h_t$, which is passed to a fully connected output layer.

$$\hat{y} = W \cdot \hat{h} + b \quad (10)$$

Then the softmax is operated to normalize the output vector $\hat{y}$ to the probability distribution as follows:

$$P(y_k = 1) = \frac{\exp(\hat{y}_k)}{\sum_j \exp(\hat{y}_j)} \quad (11)$$

where k denotes the index of the multi-hot vector y, which represents the pairs of speech act attribute and category information of utterance.

3.4. Multi-label Threshold Predictor

The output probability distribution $p(y|u)$ from the softmax layer is used for multi-label prediction, while the proposed model is trained and used in prediction for a given utterance u. A relevant label set Y for an utterance u is determined by a threshold t as follows:

$$Y = \{k | p_k > t,\ k \in L\}. \quad (12).$$

The threshold learning mechanism used in the literature [17, 18] is adopted in [D] of Figure 1, which models t with a linear regression model. The learning procedure is described as follows: For each training example $(u_m,\ Y_m)$, we set the target threshold value t_m which minimizes the count of misclassified labels as follows:

$$t_m = \underset{t}{\operatorname{argmin}}(|\{k | k \in Y_m, p_k^m \le t\}| + |\{l | l \in \overline{Y_m}, p_l^m \ge t\}|) \quad (13)$$

where p_k^m, p_l^m is the output probability of relevant label k and irrelevant label l associated with utterance u_m respectively. The target threshold values t_m is used in learning the θ of the threshold predictor $T(\boldsymbol{u}; \theta)$:

$$E(\theta) = \frac{1}{2}\sum_{m=1}^{M}(T(\boldsymbol{x}_m; \theta) - t_m)^2 + \frac{\lambda}{2}\|\theta\|^2, \quad (14)$$

where λ is the regularization parameter and M is the number of utterances in the train set. At the test time, the learned threshold is used to choose the relevant labels Y of a test utterance, as illustrated in (14).

4. Experimental Setup

In this section we introduce the DSTC5 dataset and describe the different models with their corresponding performances.

4.1. Statistics of DSTC5 Datasets

The summary statistics of the SLU datasets for the both speakers of the DSTC5 after tokenization are given in Table 2. For the case of *Guide*, one interesting point to note is that the size of sets of labels in the train set is smaller than that in the test set , which means that there is no way for the classifier to learn cases of certain labels assigned to utterances during the training and predict correct speech acts on the test dataset of *Guide*.

4.2. Hyper-parameters

In our experiments, we use: filter windows of 2, 3, 4 with 200 feature maps for the CNN, dimension of 100 for the hidden unit of LSTM. Those values are chosen by adopting a rough grid search[8]. The model undergoes training through stochastic gradient descent over shuffled mini-batches with Adam optimization algorithm. The model stops the iterant processes of learning by an early stopping mechanism.

As we mentioned in Section 3.2 that the length of dialogue context *L* is treated as a free parameter, the optimal value of *L* is

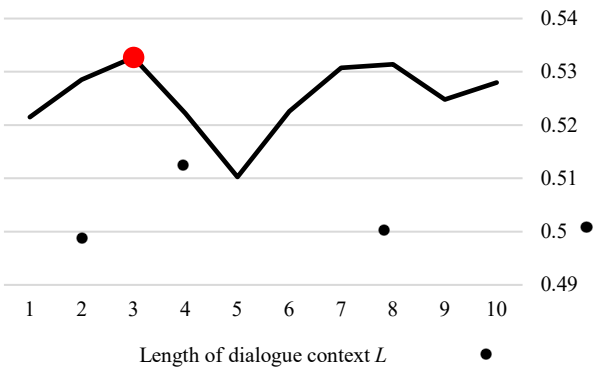

Figure 2. The performance of CNN-LSTM model based on various lengths of dialogue context on Tourist dataset.

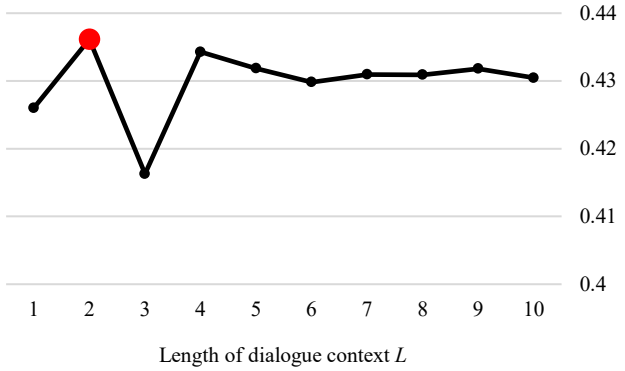

Figure 3. The performance of CNN-LSTM model based on various lengths of dialogue context on Guide dataset.

derived by conducting a grid search, where L ranges from 1 to 10 on both *Guide* and *Tourist* datasets. Figure 2 and Figure 3 illustrate the performance changes with respect to the length of dialogue context in the *Guide* and *Tourist* dataset respectively. It is shown that the optimal length of dialogue context would be L=3 and L=2 for the *Tourist* and *Guide* dataset, respectively, for making a dialogue context that improves the classification performance.

4.3. Word Embedding Vectors

GloVe[19] and *Word2Vec*[20] are the two most popular word embedding algorithms aiming at mapping semantic meaning of words in a geometric space. We initialize our proposed models with two publicly available pre-trained word vectors and both word embedding vectors have dimensionality of 300; *GloVe* that are trained on 6 billion words from Wikipedia 2014 and Gigaword5[2] and *Word2Vec* that are trained on 100 billion words from Google News[3]. In the preliminary experiments, it is observed that all the proposed models trained on top of the pre-trained *Word2Vec* show slightly better performances over those on top of pre-trained *GloVe*. In this sense only the performances of *Word2Vec* based models will be presented in this paper.

4.4. Model Variations

We evaluate three models with different architecture:

- CNN (multiclass): the model that predicts only one speech act category and attribute for given a *speaker's* utterance.
- CNN-LSTM (multiclass): the combined model which exploits dialogue context information from previous *speaker's* utterances
- CNN-LSTM (thresholding): the combined model with a threshold predictor that classifies multiple labels of speech act categories and attributes.

4.5. Evaluation Metrics

In the SLU task, a system is required to match relevant speech acts for a given unlabeled utterance spoken by the target role speaker. The following evaluation metrics are used in DSTC5.

- Precision: Fraction of speech act labels that are correctly predicted.
- Recall: Fraction of speech act labels in the gold standard that are correctly predicted.
- F-measure: The harmonic mean between precision and recall.

5. Results

Table 3 and Table 4 summarize the comparative experimental results of our models for classifying speech acts categories and attributes on *Guide* and *Tourist* datasets, respectively. On the *Tourist* dataset, the CNN-LSTM (multi-class) model shows promising results. Note that CNN-LSTM (multi-class) models outperform most of the models submitted in the DSTC5 for both speakers even without utilizing threshold learning mechanism. In terms of the F1-score, the model CNN-LSTM (thresholding) significantly outperforms all the other models on *Tourist* dataset. For the case of *Guide* dataset, our model is slightly behind of Team 2's model, but still highly comparable.

These results suggest that for conducting the SLU task of DSTC5 the CNN is a suitable model to predict and fill the slots of speech act categories and attributes. A threshold predictor enables the CNN models to classify a set of multiple labels on each utterances. It is worthy nothing that our model uses only publicly available word-embedding vectors without having any manually designed features or using delexicalization. Last but the most important thing is the dialogue context information encoded with the LSTM helps to improve the performance of conducting SLU task.

6. Discussion

At this point we circle back to the very beginning in order to understand what limitation had been posed on Xu et al.'s model and how our CNN-RNN model could tackle those issues[4]. The basic ideas of our study start from the properties of utterances. Each utterance is a sub-part of a dialogue and inevitably depends on the previous utterances in the intertwined way. In order to capture the semantic meaning from utterances the semantic

[2] http://nlp.stanford.edu/projects/glove

[3] https://code.google.com/archive/p/word2vec

[4] A reviewer suggested that the problem statement of previous model should be addressed. We thank a reviewer for bringing the comparison between previous and current models to our attention.

Table 3. Comparative results for *Tourist* dataset

Model	Precision	Recall	F1-measure
Baseline (SVM)	0.3694	0.1828	0.2446
Team 2 (LC-MTL)	0.5331	0.5263	0.5297
Team 3	0.4591	0.4241	0.4409
Team 5	0.5026	0.4484	0.4739
Team 7	0.5079	0.4156	0.4571
Xu (2017)	0.5010	0.5624	0.5299
CNN (multi-class)	**0.5462**	**0.4873**	**0.5151**
CNN-LSTM (multi-class)	**0.5603**	**0.4999**	**0.5284**
CNN-LSTM (thresholding)	**0.5455**	**0.5276**	**0.5364**

Table 4. Comparative results for *Guide* dataset

Model	Precision	Recall	F1-measure
Baseline (SVM)	0.4588	0.2480	0.3219
Team 2 (LC-MTL)	0.5127	0.4251	0.4648
Team 3	0.4340	0.3635	0.3956
Team 5	0.4639	0.3820	0.4190
Team 7	0.5007	0.2976	0.3733
Xu (2017)	0.4239	0.4295	0.4266
CNN (multi-class)	**0.4768**	**0.3927**	**0.4307**
CNN-LSTM (multi-class)	**0.4837**	**0.3983**	**0.4369**
CNN-LSTM (thresholding)	**0.4630**	**0.4256**	**0.4424**

Table 5. An example of test utterances annotated by different models.

No.	Utterances	Speech Act Category (Attribute)	
		CNN	**CNN-LSTM**
1	嗯。 uh-huh .	FOL (INFO)	FOL (ACK)
2	嗯。 uh-huh .	FOL (ACK)	RES (CONFIRM)
3	我说她们在新加坡的主要交通工具是什么? i said , what 's the main means of transport them in Singapore?	QST (WHAT)	QST (INFO)
4	政府 国家给你们出的吗? the government will give you out of the country?	QST (INFO)	QST (CONFIRM)
5	我们这样谈话的。 we talk like this.	FOL (PREFERENCE)	FOL (INFO)
6	啊，不贵 ah , it 's not expensive.	FOL (INFO)	FOL (POSITIVE)

decoder must be robust enough (i) to detect the semantic meaning from utterances regardless of the specific word order – *propositional meaning,* and (ii) to understand the intention when a speaker utters them in the middle of dialogues – *pragmatic information*.

With these preliminaries in place Xu et al.'s model, which is grounded on CNN, overlooked the importance of pragmatic information, although it has specialized capability of extracting out of the necessary information from the current utterance. Our model has concatenated RNN to CNN before the soft-max operation so that it can retrieve the contextual meaning from the previous utterances in the dialogues and build more complete concept structure.

As illustrated in Table 5, our proposed CNN-LSTM model could correctly predict the labels which are supposed to be annotated. Consider the example of an utterance 嗯 '*uh-huh'*. If we only have a look this utterance itself, there is no way to disambiguate the meaning between acknowledge and confirmation. In the case of the example 4, since it has already uttered in the previous turn and both two participants of this dialogue know this fact, the confirmational phrase such as 'you meant … is it right?' is omitted to avoid the redundancy. This pragmatic information is stored in the long-term memory and utilized to understand the genuine meaning of corresponding utterance.

7. Conclusion

This paper has extended the work of Xu et al. on the SLU task of DSTC5, and has presented a semantic decoder using deep neural networks. We have compared different models for combining a threshold predictor and long short-term memory. Our concise model proved its efficiency by outperforming the models including the baseline on the DSTC5 and those submitted in the challenge. We observed that understanding the semantic meaning and building a concept structure of a certain utterance is effectively obtained by utilizing both current and previous information. It addresses that the architecture of our concatenated CNN-LSTM model may be applicable to other types of dialogues corpora for achieving SLU.

Acknowledgement

This work was supported by the National Research Council of Science & Technology (NST) grant by the Korea government (MSIP) (No. CRC-15-04-KIST). We sincerely thank two anonymous reviewers and external readers, and the audiences of 2017 Bigcomp conference for their detailed comments. All errors and misrepresentations, if any, are solely ours.

References

[1] G. Xu, H. Lee, M. Koo, J. Seo, "Convolutional Neural Network using a Threshold Predictor for Multi-label Speech Act Classification" in Big Data and Smart Computing (BigComp), 2017 IEEE International Conference, pp. 126-130, 2017.

[2] S. Kim, L. F. D'Haro, R. E. Banchs, J. D. Williams, M. Henderson, K. Yoshino, "The Fifth Dialog State Tracking Challenge" in Proceedings of the 2016 IEEE Workshop on Spoken Language Technology (SLT), pp. 511-517, 2016.

[3] S. Kim, L. F. D'Haro, R. E. Banchs, J. D. Williams M. Henderson, K. Yoshino, Dialog State Tracking Challenge 5 handbook v3.0, 2016.

[4] T. Ushio, H. Shi, M. Endo, K. Yamagami, N. Horii, "Recurrent Convolutional Neural Networks for Structured Speech Act Tagging" in Proceedings of the 2016 IEEE Workshop on Spoken Language Technology (SLT), pp. 518-524, 2016.

[5] Y. Kim, "Convolutional Neural Networks for Sentence Classification" in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1746-1751, 2014.

[6] P. Wang, J. Xu, B. Xu, C.L. Liu, H. Zhang, F. Wang and H. Hao, "Semantic Clustering and Convolutional Neural Network for Short Text Categorization" in Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics(ACL), pp. 352-357, 2015.

[7] R. Johnson, and T. Zhang, "Semi-supervised convolutional neural networks for text categorization via region embedding" In Advances in neural information processing systems, pp. 919-927, 2015.

[8] Y. Zhang, B. C. Wallace, "A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification" in arXiv preprint arXiv:1510.03820, 2016.

[9] T. Mikolov, M. Karafiát, L. Burget, J. Cernocký and S. Khudanpur, "Recurrent Neural Network based Language Model", In Proceedings of Interspeech, 2010.

[10] M. Sundermeyer, R. Schluter, and H. Ney, "LSTM neural networks for language modeling", In Proceedings of Interspeech, 2010.

[11] Z. Huang, W. Xu, and K. Yu, "Bidirectional LSTM-CRF models for sequence tagging", In arXiv preprint arXiv:1508.01991, 2015.

[12] I. Sutskever, O. Vinyals and Q. V. Le, "Sequence to sequence learning with neural networks", NIPS Technical report, 2014.

[13] O. Vinyals, A. Toshev, S. Bengio, D. Erhan, "Show and tell: Lessions learned from the 2015 mscoco image captioning challenge" IEEE transactions on pattern analysis and machine intelligence, **39**(4), 652-663, 2017.

[14] S. Kim, R. E. Banchs, H. Li, "Exploring Convolutional and Recurrent Neural Networks in Sequential Labelling for Dialogue Topic Tracking" in Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL), pp. 963-973, 2016.

[15] L. M. R. Barahona, M. Gasic, N. Mrkšić, P. H. Su, S. Ultes, T. H. Wen, S. Young, "Exploring Sentence and Context Representations in Deep Neural Models for Spoken Language Understanding" in Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pp. 258-267, 2016.

[16] S. Hochreiter, J. Schmidhuber, "Long short-term memory" Neural computation, **9**(8), 1735-1780, 1997.

[17] A. Elisseeff, J. Weston, "A kernel method for multi-labelled classification" Advances in Neural Information Processing Systems, **14**, 681-687, 2001.

[18] J. Nam, J. Kim, E. L. Mencía, I. Gurevych, and J. Furnkranz, "Large-scale multi-label text classification – revisiting neural networks" Machine Learning and Knowledge Discovery in Databases, Springer, 437-452, 2014.

[19] P. Jeffery, R. Socher, C. D. Manning, "Glove: Global vectors for word representation" in Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP), vol. 14, pp. 1532-1543, 2014.

[20] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean, "Distributed representations of words and phrases and their compositionality" Advances in Neural Information Processing Systems, 3111-3119, 2013.