Toward Multi-lingual Natural Language Understanding using Deep Neural Networks

Abstract

Recent approaches based on deep neural networks have shown promising results for natural language understanding. However, those models are evaluated on monolingual corpus. In this work, we present a semantic decoder based on convolutional neural network and long-short term memory network to exploit a current user's utterance and previous system's utterances. It is shown that our model is multilinguistically applicable by classifying a set of dialogue act-slot pairs on both corpora built in Korean and English.

1 Introduction

Natural language understanding (NLU) has been one of the most rudimentary components in human-machine conversation (Wang et al., 2015). In order to achieve NLU, there is a need to capture pragmatic intention and to extract semantic meanings from an almost infinite variety of a user's utterances in the middle of dialogues. In the light of this situation, a semantic decoder should be able to consider both the current user's utterances and the previous conversation. Although Rojas-Barahona et al. (2016) proposes a joint model that is capable of capturing both sentence- and context- representation, their work is only evaluated on English corpora (i.e., DSTC2 and in-car datasets). This naturally raises a question whether this joint model could be extended across different languages.

The aims of this study are two folds: first, to build a robust semantic decoder by concatenating two deep neural networks to exploit multiple inputs, and secondly to conduct a dialogue act-slot pairs classification task on two corpora – one for Korean (i.e., SGDSG[1]) and the other for English dialogues (i.e., DSTC2). To fulfill this objective, we will briefly review previous studies of NLU in section 2. Section 3 will present the details of the architecture of our concatenated model, and section 4 will summarize the experimental set up. Section 5 will provide the experiment result that shows the robustness of our semantic decoder. In Section 6, we will conclude the paper and remarks on main findings of our study with implication of our future research.

2 Related Works

With the development of deep learning, typical deep learning models such as convolutional neural networks (CNN) and recurrent neural networks (RNN) have achieved remarkable results in several natural language processing tasks (Kim 2014; Mikolov 2010; Socher et al., 2012). Recently, several researches have been conducted by combining CNN and RNN models. Kim (2016) proposes a recurrent convolutional network (RCNN) model, in which the penultimate layer of CNN is connected to the recurrent layers in the RNN model to track a topic of a dialogue in human-human conversations.

Another jointed CNN and RNN model proposed by Rojas-Barahona et al. (2016) is different from previous CNN-RNN models, in that they optimize the model with two distinctive inputs: a current user' utterance and act-slot pairs of previous system utterances. In the task of decoding semantic meaning of spoken languages each input is utilized in sentence representation and context representation, respectively.

[1] Sogang Dialogue System Group.

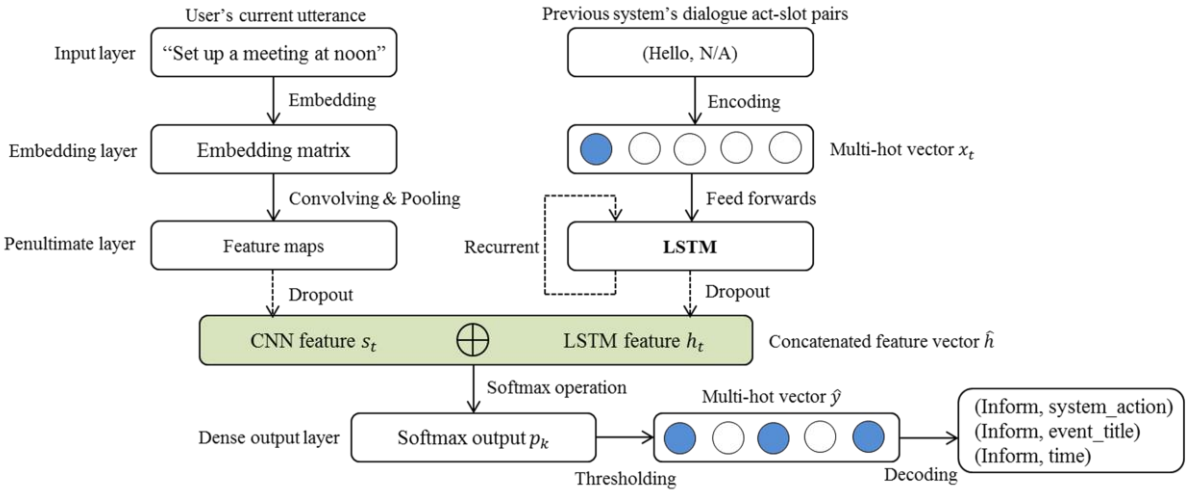

Figure 1. The Architecture of concatenated CNN-LSTM model.

3 Models

In this section, we will introduce the architecture of our semantic decoder, as illustrated in Figure 1.

3.1 Convolutional Neural Network

Our model for predicting a correct set of dialogue act-slot pairs from a corresponding user utterance is based on the CNN architecture proposed by Collobert et al. (2011) and Kim (2014). In this architecture, a sentence of length n is represented as a $n \times k$ matrix. Each row of the matrix is a k dimensional morpheme embedding vector $x_i \in \mathbb{R}^k$ representing the i-th word in a sentence. Each word in a sentence is segmented into several morphemes by *Komoran* [2] (Park and Cho, 2014), which are initialized into embedding vectors for an input layer.

A convolutional operation involves a filter $\boldsymbol{m} \in \mathbb{R}^{h \times k}$ is applied to a window of h rows to produce a feature:

$$c_i = f(\boldsymbol{m} \cdot x_{i:i+h-1} + b)\,, \qquad (1)$$

where f is a hyperbolic tangent function and $b \in \mathbb{R}$ is a bias term. These series of convolutional operations are applied to all the possible windows and generate a feature map:

$$\boldsymbol{c} = [c_1, c_2, \ldots, c_{n-h+1}]\,. \qquad (2)$$

Then a max pooling is operated to take the maximum value $\hat{c} = \max\{\boldsymbol{c}\}$ as a representative feature for the filter.

In our model, multiple filters with varying window size h are integrally engaged to obtain multiple adjacent features. These features are then concatenated to form the 'top-level' feature vector s_t, which embeds features of a user utterance at a dialogue turn t.

3.2 Long-Short Term Memory Network

Since each utterance is dependent on the previous utterances in a conversation, it is necessary to refer to dialogue act-slot instances of a system's utterance right before a user says. To receive assistance from the previous system's utterances at an inter-utterance level as well, we employ a long-short term memory (LSTM) (Hochreiter and Schmidhuber, 1997), a special kind of recurrent neural network (RNN) which is much better for preserving information over long periods of time than other kinds of traditional RNN.

The structure of LSTM is divided into a memory cell c_t and three gates: a forget gate f_t, an input gate i_t and an output gate o_t. Three kinds of gates functions to decide which amount of information the memory cell should keep or forget at a time step t. The input x_t and the output h_t of LSTM are updated as follows:

$$i_t = \sigma(W^i \cdot x_t + U^i \cdot h_{t-1} + b^i) \qquad (3)$$
$$f_t = \sigma(W^f \cdot x_t + U^f \cdot h_{t-1} + b^f) \qquad (4)$$
$$o_t = \sigma(W^o \cdot x_t + U^o \cdot h_{t-1} + b^o) \qquad (5)$$
$$g_t = tanh(W^g \cdot x_t + U^g \cdot h_{t-1} + b^g) \qquad (6)$$
$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \qquad (7)$$
$$h_t = o_t \odot tanh(c_t) \qquad (8)$$

where x_t is the input at the current time step, h_t is the hidden unit at time step t, b is a bias term, $\sigma(\cdot)$ is a logistic sigmoid function and $\odot$ denotes a point-wise multiplication operation.

[2] Though *Komoran* is one of Korean POS-taggers, we use it to segment the words in an utterance, and do not encode any morphosyntactic information tagged to each morpheme.

Corpus	Speaker	Utterance	Act	Slot	Value
SGDSG	System	무엇을 도와드릴까요? *mwuesul towatulilkkayo*? "How can I help you?"	Hello	N/A	N/A
	User	나 내일 회의일정 등록해줘. *na nayil hoyuyilceng tunglokhaycwu.* "Schedule a tomorrow's meeting."	Inform Inform Inform	SYSTEM_ACTION DATE EVENT_TITLE	create tomorrow meeting
DSTC2	System	What kind of food would you like?	Request	FOOD	N/A
	User	Cheap Indian food	Inform Inform	FOOD PRICE_RANGE	Indian cheap

Table 1: Example of a user's and system's utterances annotated with dialogue act-slot-value pairs.

Unlike the model proposed by Rojas-Barahona et al. (2016), where the word vectors are fed into LSTM as inputs, we encode corresponding dialog act-slot pairs of previous system's utterances at a time step t into a single multi-hot vector, and stipulate them as the input x_t. As the single multi-hot vector represents multiple labels of dialogue act-slot pairs as a whole and is fed into LSTM at a time *t*, it facilitates the network to grasp semantic information at an inter-utterance level more straightforward.

3.3 Concatenating CNN and LSTM

We notice that the contextual flow of a conversation works as just an auxiliary information in extracting the semantic meaning from the current user's utterances. While feature vectors in the penultimate layer of CNN and RNN are merged by a tangent function in Rojas-Barahona et al. (2016), we concatenate the hidden unit h_t of LSTM to the 'top-level' feature vector s_t modeled by the CNN. Then, the concatenated vector $\hat{h}_t = s_t \oplus h_t$ is passed to a fully connected softmax layer whose output is the probability distribution over all labels of dialogue act-slot pairs as described in Figure 1.
The softmax operation over each prediction is calculated as follows:

$$P(y_k = 1|\hat{h}, W, b) = \frac{\exp(W_k \cdot \hat{h} + b_k)}{\sum_j \exp(W_j \cdot \hat{h} + b_j)} \quad (8)$$

where k denotes the index of the multi-hot vector y, which represents the dialogue act-slot pairs of user's utterance.

3.4 A Threshold Predictor

In both datasets of SGDSG and DSTC2, more than one label of a dialogue act-slot pair is annotated to a given single user's utterance. To perform this multi-label classification task, we use the output probability distribution p_k for a given utterance $\boldsymbol{x}$ from the softmax layer. The predicted multiple label of act-slot pairs $\hat{y}$ for an utterance $\boldsymbol{x}$, is determined by a threshold t as follows:

$$\hat{y} = \{y_k | p_k > t;\ k \in L\} \quad (9)$$

The threshold learning mechanism used in the literature (Elisseeff and Weston, 2001; Nam et al., 2014) is adopted, which models t with a linear regression model.

4 Experimental Setup

4.1 Corpus Development

Evaluation of model is conducted on two datasets: SGDSG in Korean and DSTC2 (Henderson et al., 2014b) in English. Both corpora are annotated within the tagging framework (i.e., Dialogue act-slot-value triplets) of the DSTC2, as illustrated in Table 1. Since there has been existed no dialogue corpus in Korean, building a dialogue corpus with transcribed texts is the highest priority task in our research. The SGDSG corpus is collected on the topic of schedule management. The user is able to create, read, update, and delete schedules. In each dialogue, the details such as start date, alert, event title, location are required for the system to access to or update database.

Corpus	U	D	L	C
SGDSG	6,480	1,529	33	1.48
DSTC2_train	11,677	3,934	101	1.23
DSTC2_test	1,612	506	76	1.20

Table 2: Statistics of SGDSG and DSTC2 corpus. **U**: Number of utterances spoken by a user. **D**: Number of total dialogues. **L**: Size of all possible dialogue act-slot pairs in the corpus. **C**: Average number of dialogue act-slot pairs tagged per utterance.

The DSTC2 corpus is collected for the purpose of providing restaurant information in the city of Cambridge. The system searches a restau-

Model	Precision	Recall	F1-measure
CNN (multiclass)	73.43 ±0.32	49.54 ±0.22	59.16 ±0.26
CNN (threshold)	92.83 ±0.99	90.77 ±1.14	91.74 ±1.04
CNN-LSTM (threshold)	94.76 ±0.77	94.45 ±0.94	94.58 ±0.50

Table 3: Evaluation of our models on the SGDSG corpus.

rant by three constraints such as area, price range, and food type, and after specifying a certain place a user is able to query the system for other information such as address and phone number.

4.2 Hyper-parameters and Training

In our experiments, we use: filter windows (h) of 2, 3, 4 with 200 feature maps each for the CNN, dimension of 128 for the hidden unit of LSTM and a batch size of 60. As a means of regularization, we apply Dropout on the penultimate layers of both the CNN and the LSTM with dropout rate of 0.2. Those values are chosen by performing a rough grid search (Zhang and Wallace, 2016). The model undergoes training through stochastic gradient descent over shuffled mini-batches with RMSprop update rule. Our model stops the iterant processes of learning by an early stopping mechanism.

4.3 Model Variations

To evaluate the classification performance of our CNN-LSTM combined model, we compare the performance of three models[3]:

- CNN (multiclass): The model that predicts only one dialogue act-slot for given user's utterance.
- CNN (threshold): The CNN model with a threshold predictor that classifies multiple labels of dialogue act-slot pairs
- CNN-LSTM (threshold): The concatenated CNN and LSTM model which allows to extract information from both a current user's and previous system' utterances.

5 Results and Discussion

As for the SGDSG corpus, we conduct a 5-fold cross validation task. All models iterate the evaluation process 20 times, and the mean scores and standard deviations of each evaluation metric are calculated. Table 3 summarizes the comparative results of each model on the SGDSG corpus. It is observed that the performance of the CNN model is significantly improved with the help of a threshold predictor, which enables the semantic decoder to predict multiple pairs of dialogue act-slot. Further improvements are achieved by CNN-LSTM model. It points that this conjoined model more effectively capture the semantic meaning by utilizing multiple inputs: a current user's utterance and previous system's utterances.

Model	Precision	Recall	F1-measure
CNN (Cambridge)	89.73	84.74	87.14
CNN-LSTM_w4 (Cambridge)	88.95	86.02	87.43
CNN (threshold)	89.29	83.70	86.40
CNN-LSTM (threshold)	88.34	85.96	87.18

Table 4: Comparative results of different models on the DSTC2 corpus.

We also evaluate our models on the DSTC2 corpus built in English, whose experimental results are shown in Table 4. We observe that the concatenated CNN and LSTM model still maintain the desirable performance on English corpus as well, compared to the best-performing model on the DSTC2 corpus (Rojas-Barahona et al., 2016). The results suggest that our CNN-LSTM model is solid and robust enough to conduct a NLU task regardless of what language the corpus is built in. It is worth noting that though Korean and English are morphologically far different, our CNN-LSTM model steadily predicts correct label set of dialogue act-slot pairs well, without using any manually designed feature or preprocessing the data through delexicalisation.

6 Conclusion and Discussion

In this paper we have presented a CNN-LSTM based approach to conduct a NLU task. We demonstrate that concatenating two networks facilitates a system to classify a correct set of labels for a given utterance, by using inputs at both utterance and inter-utterance level. Our model achieves outstanding results on multi-lingual corpora of dialogues. We will extend our research to decode slot-value pairs in future research.

[3] Codes are available at https://github.com/hkhpub/cnn-lstm-slu .

References

Lina M. Rojas Barahona, Milica Gasic, Nikola Mrkšić, Pei-Hao Su, Stefan Ultes, Tsung-Hsien Wen, and Steve Young. 2016. Exploiting 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*, pages 258–267.

Ronan Collobert, Jason Weston, L´eon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537.

Andre Elisseeff and Jason Weston. 2001. A kernel method for Multi-labelled classification. In *Advances in Neural Information Processing Systems (NIPS), Volume 14*, pages 681-687.

Matthew Henderson, Blaise Thomson, and Jason Williams. 2014. The second dialog state tracking challenge. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 263-272.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long shor-term memory. *Neural computation, Volume 9, Issue 8*, pages 1735–1780.

Minwoo Jeong and Gary Geunbae Lee. 2006. Exploiting Non-local Features for Spoken Language Understanding. In *Proceedings of the COLING/ACL 2006 Main Conference*, pages 412–419.

Seokhwan Kim, Rafael E. Banchs, and Haizhou Li. 2016. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics,* pages 963–973.

Yoon Kim. 2014. Convolutional Neural Networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751.

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernocký, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Proceedings of Interspeech/ICSA*, pages 1045–1048.

Jinseok Nam, Jungi Kim, Eneldo Loza Mencía, Iryna Gurevych, and Johannes Fürnkranz. 2014. Large-scale Multi-label Text Classification - Revisiting Neural Networks. *Machine Learning and Knowledge Discovery in Databases, Springer*, pages 437–452.

Eunjeong L. Park and Sungzoon Cho. 2014. KoNLPy: Korean natural language processing in Python. In *Proceedings of the 26th Annual Conference on Human & Cognitive Language Technology*.

Richard Socher, Alex Perelygin, Jean Y. Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing(EMNLP)*, pages 1631–1642.

Ye-Yi Wang, Li Deng, and Alex Acero. 2005. Spoken language understanding. *IEEE Signal Processing Magazine*, pages 16–31.

Ye Zhang and Byron C. Wallace. 2016. A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification. *arXiv preprint arXiv:1510.03820.*