# NEURAL DIALOG STATE TRACKER FOR LARGE ONTOLOGIES BY ATTENTION MECHANISM

*Youngsoo Jang*, Jiyeon Ham*, Byung-Jun Lee, Youngjae Chang, Kee-Eung Kim*

School of Computing
KAIST
Daejeon, South Korea

## ABSTRACT

This paper presents a dialog state tracker submitted to Dialog State Tracking Challenge 5 (DSTC 5) with details. To tackle the challenging cross-language human-human dialog state tracking task with limited training data, we propose a tracker that focuses on words with meaningful context based on attention mechanism and bi-directional long short term memory (LSTM). The vocabulary including a plenty of proper nouns is vectorized with a sufficient amount of related texts crawled from web to learn a good embedding for words not existent in training dialogs. Despite its simplicity, our proposed tracker succeeded to achieve high accuracy without sophisticated pre- and post-processing.

***Index Terms*—** Recurrent Neural Network, Dialog state tracking, DSTC5, Attention mechanism, Word embedding

## 1. INTRODUCTION

The dialog state tracking challenge (DSTC) is a research challenge of recognizing the intentions of the users from noisy utterances in task-oriented dialogs. Unlike the previous challenges (DSTC 2&3) where the human-system dialogs were given, provided dialogs are replaced with human-human dialogs in DSTC 4&5 widening the diversity of expressions. The cross-language feature is further added in DSTC 5, where the source language and the target language is now different, becoming the most challenging task so far [1].

Among numerous methodologies, recurrent neural network (RNN) and its variants are now common in handling sequential data with their promise of performance and ease of use. Long-short term memory (LSTM) [2], one of the most popular RNN variants that is able to capture long-range dependencies, is commonly adopted. Their performance has shown great potential in previous challenges (DSTC 2,3&4) ([3], [4], [5]). Here, we adopted the bi-directional LSTM, which has been found to outperform other models in sequential data processing that requires capturing local context ([6], [7]).

| **Topic** | ATTRACTION |
|---|---|
| **Utterance** | Uh what about East Coast Beach? |
| | East Coast Beach is also a nice place. |
| | East Coast? |
| | Yah. |
| | Uh East Coast. |
| **Label** | INFO: Preference<br>PLACE: East Coast Park<br>TYPE_OF_PLACE: Beach |

**Table 1**. Example of training set segment in DSTC 5. Given **Topic** and **Utterance**, we should predict the slot-value pairs in **Label**. INFO, PLACE and TYPE_OF_PLACE are the examples of slots, and Preference, East Coast Park and Beach are corresponding values of such slots.

The main difficulties in using such models are, however, the large size of the ontology and sparsity of the training data. Since only 56% of keywords in the ontology are present in the training dialogs, it is not possible to completely train typical models and the need of special treatment arises (e.g. value independent network in [3]).

In this paper, we present a tracker based on bi-directional LSTM with attention mechanism, which is the recent trend in deep learning. After eliminating obvious stop words and reducing lexical variations, pre-trained word embedding model converts linguistic words into vectors. Bi-directional LSTM works on word embedding vectors to find notable keywords in each utterance. Then the tracker finds proper values for each slot from weighted sum of word vectors with attention weight which is determined from the bi-directional LSTM. Finally, the tracker decides whether the found values are reliable based on entropy or cosine similarity.

This paper is organized as follows. In Section 2, we give a description of the DSTC 5 dataset and the main task, with the brief review of previous works of LSTM and attention mechanism. In Section 3, we explain our dialog state tracker

*:These authors contributed equally. This material is based upon work supported by the Ministry of Trade, Industry & Energy (MOTIE, Korea) under Industrial Technology Innovation Program. No.10063424, 'Development of Distant Speech Recognition and Multi-Task Dialog Processing Technologies for In-Door Conversational Robots'.

| | POS | Description | POS | Description |
|---|---|---|---|---|
| **maintained** | VB | base form verb: ask, assume, build | NN | singular noun: child, dog, car |
| | VBD | past tense verb: adopted, used, asked | NNS | plural noun: children, dogs, cars |
| | VBG | present participle verb: using, focusing | NNP | proper noun: Singapore, Merlion |
| | VBZ | 3rd person singular verb: marks, bases | CD | numeral, cardinal: nine-thirty, 1987, one |
| | TO | 'to' as preposition or infinitive marker | IN | preposition: among, upon, on, by |
| **neglected** | RB | adverb: occasionally, very, only | CC | conjunction: and, or, but, either |
| | DT, EX, FW, LS, MD, PDT, PRP, RBR, RBS, SYM, UH, WDT, WP, WRB | | | |

**Table 2**. List of POS tags to be maintained or neglected. We maintain nouns, verbs, numerals and prepositions and neglect the others.[2]

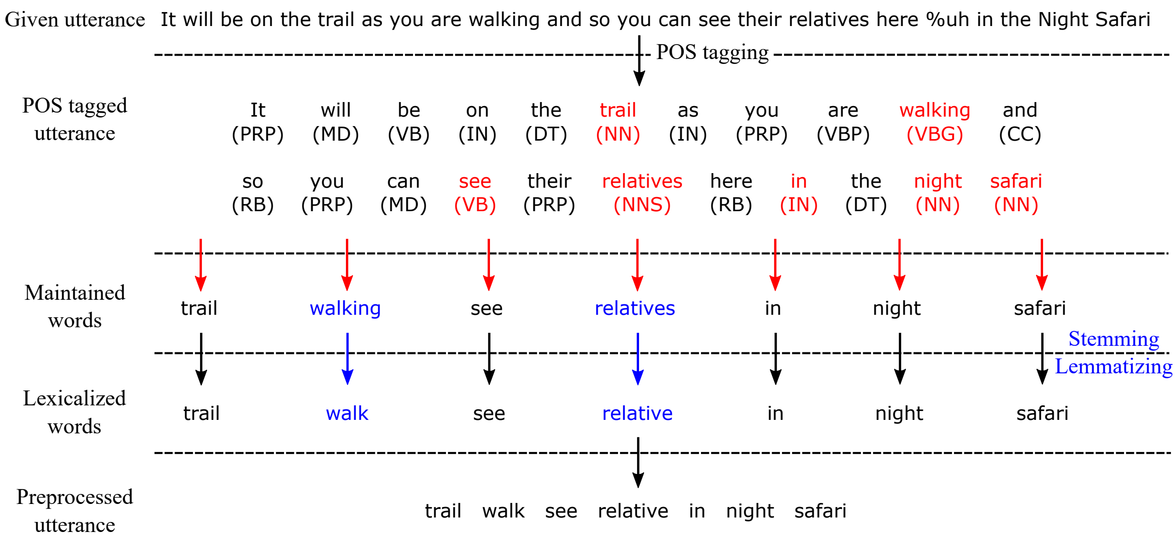



**Fig. 1**. The preprocessing example in our tracker.

part-by-part, and in Section 4, we discuss the detailed results of DSTC 5 main task.

## 2. BACKGROUND

### 2.1. Task description

In DSTC 5, the dialog state trackers compete each other over TourSG dataset. TourSG dataset consists of dialog sessions on touristic information for Singapore collected from Skype calls between a tour guide and a tourist. Dialog states are defined for each sub-dialog level, denoted as the segment. A full dialog session is divided into segments considering their topical coherence.

Each segment has one topic and dialog state, defined by the collection of slot-value pairs. The main task is to fill the slot-value pairs in each segment where the corresponding topics and utterances are given (see Table 1). Possible slot-value pairs are provided in the form of ontology. Although there exist some cases that more than one value are assigned in single slot, we only predicted one value per slot for simplicity.

[2]see http://www.nltk.org/book/ch05.html for detailed description of POS

Slots are categorized into two types: regular slots and INFO slot. While regular slots are filled if some specific values have directly discussed in the segment, INFO slot takes the place if such specific values are not discussed, and is filled with corresponding subject.

DSTC 5 differs from previous challenges in that it is a Chinese-English cross-language task. The goal of the task is to construct the dialog state tracker on Chinese dialog while the training set is given in English. The dataset consists of 35 English training set, 2 Chinese validation set and 10 Chinese test set. The top 5 results of machine translations of all dialogs are provided. In case of Chinese dialogs in validation and test set, we use the topmost machine translation result.

### 2.2. RNN and LSTM

RNN and LSTM have been generally applied to natural language processing (NLP) problems. The basic idea of RNN is to make use of sequential information. The fundamental neural network does not consider the dependency of all inputs and outputs. However, in various tasks, there exist dependency in inputs and outputs, such as sentence analysis. RNN re-

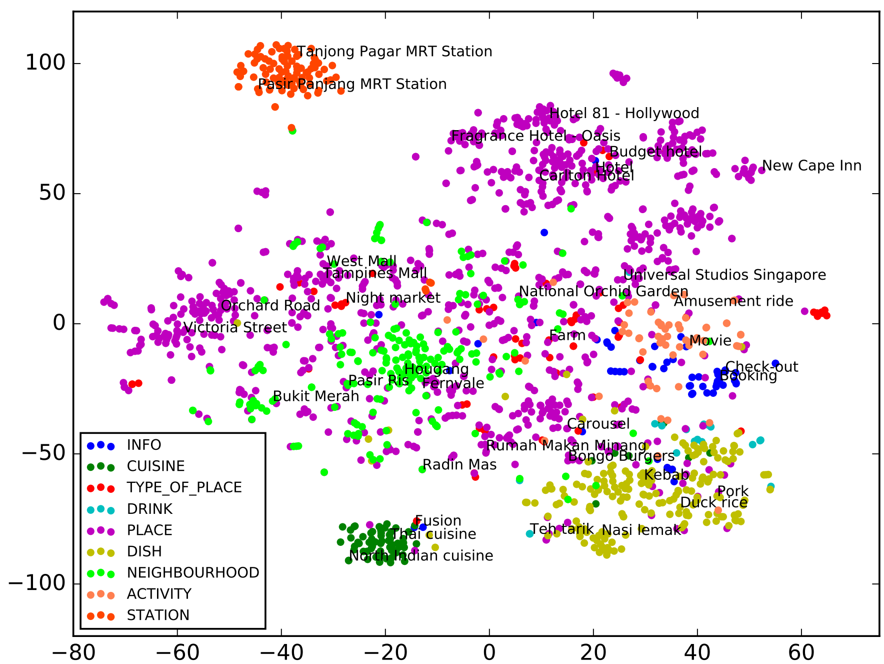



**Fig. 2**. Two-dimensional t-SNE [8] embeddings of key phrases of ontology learned by Word2Vec. The figure shows clear clusters of words of similar concepts. The phrases shown in the figure are chosen randomly.

currently use the previous computation result to compute the current output. LSTM is RNN with gates, which is proposed to prevent the vanishing gradient problem, becoming more effective in dealing with long sequences. The basic structure of LSTM unit consists of a cell state with three essential gates: input gate, forget gate and output gate. The cell controls the information storing for a long period via gates. Given an input vector $\mathbf{x}_t$ at time step $t$, the formal equation for updating gates, output and cell state are defined as follows:

$$\mathbf{i}_t = \sigma\left(\mathbf{x}_t\mathbf{U}^i + \mathbf{h}_{t-1}\mathbf{W}^i\right)$$

$$\mathbf{f}_t = \sigma\left(\mathbf{x}_t\mathbf{U}^f + \mathbf{h}_{t-1}\mathbf{W}^f\right)$$

$$\mathbf{o}_t = \sigma\left(\mathbf{x}_t\mathbf{U}^o + \mathbf{h}_{t-1}\mathbf{W}^o\right)$$

$$\mathbf{c}_t = \mathbf{c}_{t-1} \circ \mathbf{f}_t + \mathbf{i}_t \circ \tanh\left(\mathbf{x}_t\mathbf{U}^c + \mathbf{h}_{t-1}\mathbf{W}^c\right)$$

$$\mathbf{h}_t = \tanh\left(c_t\right) \circ \mathbf{o}_t$$

where $\mathbf{W}^i, \mathbf{W}^f, \mathbf{W}^o, \mathbf{W}^c \in \mathbb{R}^{N\times N}$,$\mathbf{U}^i, \mathbf{U}^f, \mathbf{U}^o, \mathbf{U}^c \in \mathbb{R}^{N\times N}$ are weight matrices, $\mathbf{h}_t$ is output vector and $i, f$ and $o$ represent input ($i$), forget ($f$) and output ($o$) gates.

### 2.3. Related works

It is now very popular to use neural networks in NLP tasks. In the last challenge (DSTC 4) that had very similar dataset to this challenge, two teams have proposed neural network based model. [9] reduced the task into multi-domain text classification problem by focusing on INFO slot filling. Using convolutional neural networks (CNN), they combined topic-shared and topic-specific structures. [5] suggested the tracker, which integrates with the baseline tracker and the unidirectional LSTM to convey the information of previous utterances. The aforementioned trackers, however, predict by outputting one-hot-encoding on the ontology, which is not easily trainable for the data with much larger ontology than the training data. More recently, Neural Belief Tracker (NBT, [10]) used deep neural network and convolutional neural network with pre-trained word embedding model.

To overcome the same problem, we take a slightly different approach—the attention mechanism. Attention mechanism is a method to focus on the meaningful information in the utterance. In dialog example, not all words in each sentence are related with state of dialog. Attention weight represents the contribution of words towards slot-specific information and it is computed using a softmax. Attention mechanism has been broadly applied to other NLP tasks such as sentence summarization [11], recognizing textual entailment(RTE) [12], and machine translation [13].

CopyNet [14] is an interesting variant of attention-based RNN Encoder-Decoder model adopting the copying mecha-

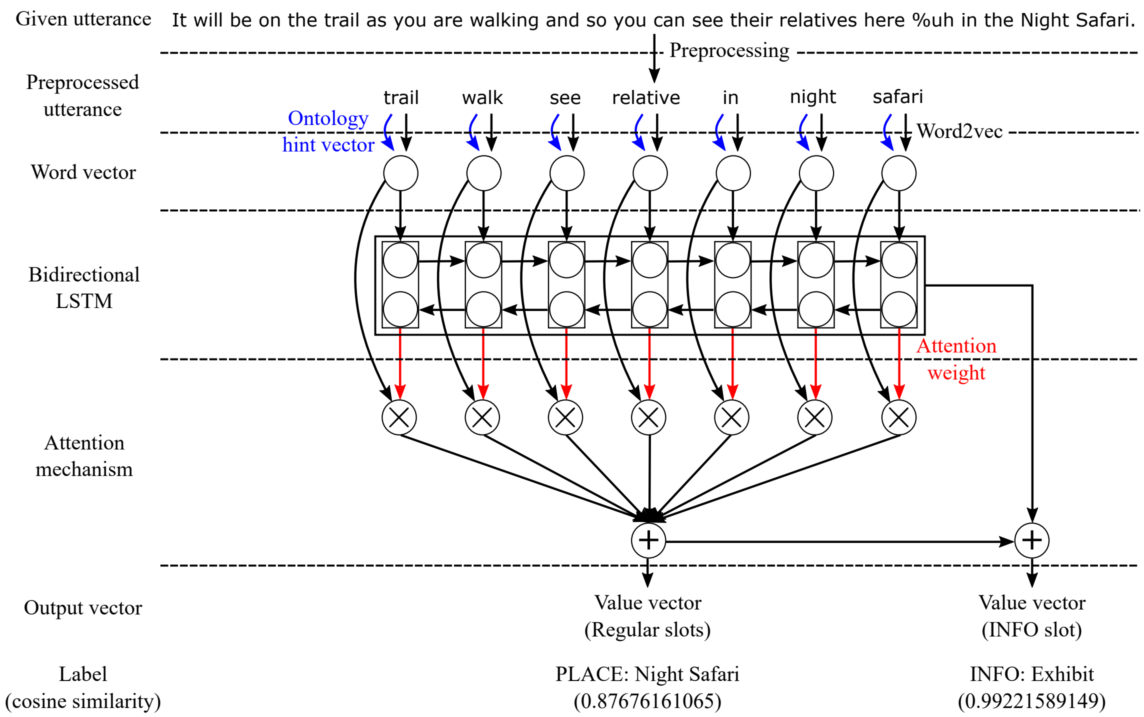



**Fig. 3**. The overall architecture of our tracker. We consider regular slots and INFO slot separately. Our tracker is inputted with the utterance, and outputs the value vectors (see Section 3 for more details).

nism to deal with large ontology. CopyNet has the ability to handle out-of-vocabulary words as proper nouns by copying consecutive sequence. For our task, target values are not long enough to be predicted sequentially. We overcome proper noun issue by improving embedding model.

## 3. TRACKER ARCHITECTURE

### 3.1. Preprocessing utterances

#### *3.1.1. POS tagging, Stemming and Lemmatization*

POS tagging is the process of classifying words and labeling them with lexical categories. TourSG corpus contains insignificant words such as articles (a, an, the), pronouns (they, we, he, she), auxiliary verbs (can, will, could, would) and onomatopoeia (uh, umm, ha). We identify and eliminate the word that does not affect the meaning of utterance by POS tagging so that the clarified utterance is easier to be understood by LSTM. We maintain nouns, verbs, adjectives, propositions and numerals, and rule the others out with the pre-trained POS tagging module in natural language toolkit (NLTK) (see Table 2).

Since the variation of words such as tense and plurality adds extra complication, there is also a need of converting these as a lexicalized form for the efficient embedding of words. We standardize the words by using porter-stemmer and word-net-lemmatizer in NLTK.

#### *3.1.2. Word embedding*

Now each utterance contains only meaningful lemmatized words. We project those words into a high-dimensional space maintaining relationship between them using Word2Vec. We convert each word into 100 dimensional embedding vector. More than 13 million sentences crawled from TripAdvisor[3] are used as training set with given TourSG corpus.

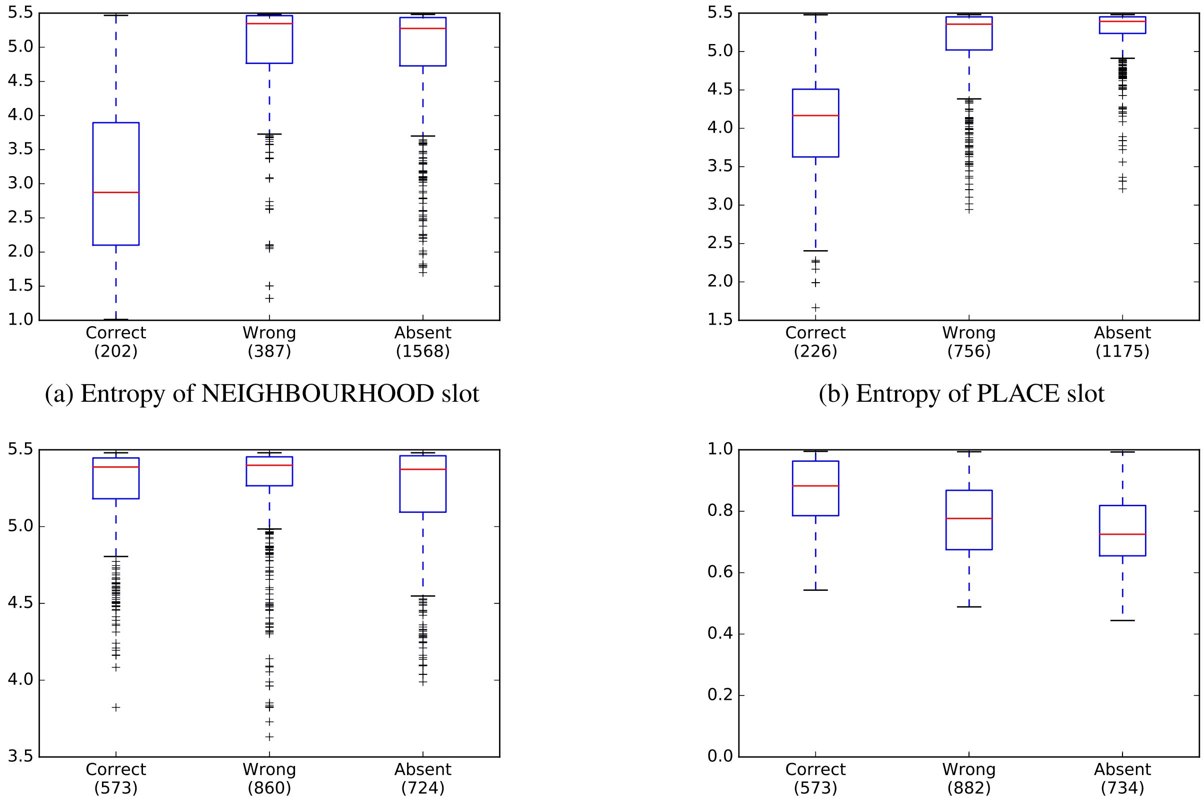
#### *3.1.3. Ontology hint vector*

Ontology also contains a valuable information about the topic to which a word is related. We construct the ontology hint vector that denotes whether a word is in the ontology or not for each topic to convey such information, resulting 30 dimensional one-hot vector. We concatenate an ontology hint vector to a word embedding vector and use it as an input for value prediction network.

### 3.2. Predicting values using attention mechanism

Our value prediction model consists of a bi-directional LSTM and a network of attention mechanism. Regard an utterance $u$ that consists of $N$ words $(u_1, u_2, \cdots, u_N)$. Each word has word embedding vector $(\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_N)$ and ontology hint vector $(\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_N)$. The input of the LSTM is a concatenated vector of a word embedding vector and an ontology hint

[3]https://www.tripadvisor.com




(a) Entropy of NEIGHBOURHOOD slot

(b) Entropy of PLACE slot

(c) Entropy of INFO slot

(d) Cosine similarity of INFO slot

**Fig. 4**. The entropy of attention weights and the cosine similarity between prediction and closest word in ontology. Correct denotes that current utterance has a relevant slot and output value is correct. Wrong denotes that output value is not correct. Absent denotes that current utterance does not have a relevant slot. The numbers in parenthesis are the counts of sample cases.

vector of each word:

$$[\mathbf{w}_1 \oplus \mathbf{c}_1; \mathbf{w}_2 \oplus \mathbf{c}_2; \cdots ; \mathbf{w}_N \oplus \mathbf{c}_N] \tag{1}$$

where $\oplus$ denotes vector concatenation. Each word has the output of the bi-directional LSTM:

$$H = [\mathbf{h}_1; \mathbf{h}_2; \cdots ; \mathbf{h}_N] \tag{2}$$

The cell values are passed to the time-distributed dense networks for each slot to formulate a single scalar per word that works as importance of corresponding word. These scalars are normalized with softmax function over words in the utterance, and further denoted as attention weights:

$$\mathbf{a}^s = \sigma(W^s H) \tag{3}$$

where $\mathbf{a}^s \in \mathbb{R}^N$ is attention weight vector, $W^s$ is weight matrix of the dense network, $\sigma$ is a softmax function, and superscript $s$ denotes a specific slot.

As an attention mechanism, we calculated the output value vector by a weighted sum of word vectors with the weights from dense network of each slot:

$$\mathbf{v}_s = \sum_{i=1}^{N} a_i^s \mathbf{w}_i \tag{4}$$

where $a_i^s$ is scalar value at index $i$ of $\mathbf{a}^s$. While attention weights are calculated for all slots, the model obtains values for all slots. This vector is now can be said to contain an essential information of the utterance. In contrast to conventional attention mechanism that utilizes a weighted sum of hidden cell vectors of LSTM, the model proposed here outputs a weighted sum of input words to ensure clear focus on key phrases.

The tracker chooses the word among the available slot values listed in ontology with the closest embedding by cosine similarity: $\mathcal{S} = \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\| \|\mathbf{b}\|}$.

### 3.3. Excluding unreliable predictions

It is equally important to decide what unreliable predictions to exclude as it is to predict correct values in dialog state tracking challenge since to predict nothing is also a choice.

| Team | Schedule 1 Accuracy | Precision | Recall | F-measure | Schedule 2 Accuracy | Precision | Recall | F-measure |
|---|---|---|---|---|---|---|---|---|
| Baseline 1 | 0.0250 | 0.1148 | 0.1102 | 0.1124 | 0.0321 | 0.1425 | 0.1500 | 0.1462 |
| Baseline 2 | 0.0161 | 0.1743 | 0.1279 | 0.1475 | 0.0222 | 0.1979 | 0.1774 | 0.1871 |
| Team 1 | 0.0417 | 0.3650 | 0.2795 | 0.3166 | 0.0612 | 0.3811 | 0.3548 | 0.3675 |
| Team 2 | 0.0788 | 0.5195 | 0.3315 | 0.4047 | 0.0956 | 0.5643 | 0.3769 | 0.4519 |
| Team 3 | 0.0351 | 0.3216 | 0.1515 | 0.2060 | 0.0505 | 0.3350 | 0.2045 | 0.2539 |
| Team 4 | 0.0583 | 0.4008 | 0.2776 | 0.3280 | 0.0765 | 0.4127 | 0.3284 | 0.3658 |
| Team 5 | 0.0330 | 0.3377 | 0.2318 | 0.2749 | 0.0520 | 0.3637 | 0.3044 | 0.3314 |
| **Ours** | **0.0491** | **0.4684** | **0.2193** | **0.2988** | **0.0643** | **0.4758** | **0.2623** | **0.3381** |
| Team 7 | 0.0092 | 0.4287 | 0.0431 | 0.0783 | 0.0107 | 0.4000 | 0.0441 | 0.0794 |
| Team 8 | 0.0192 | 0.3130 | 0.1048 | 0.1570 | 0.0214 | 0.3021 | 0.1046 | 0.1554 |
| Team 9 | 0.0231 | 0.1139 | 0.1090 | 0.1114 | 0.0314 | 0.1412 | 0.1487 | 0.1449 |

**Table 3**. The DSTC 5 main task results of the best trackers from each team, chosen based on schedule 2 accuracy.[4]

Since uncertainty estimates are not available in neural networks, other criteria had to be chosen.

#### *3.3.1. Entropy of attention weights*

For the regular slots, some keywords or phrases usually lead to the correct answers, and it is more likely to have concentrated attention weights on specific part of the utterances. The entropy of attention weights $\mathcal{H} = -\sum_{i=1}^{N} a_i^s \log a_i^s$ is an attractive choice in this sense: predictions with concentrated weights result in low entropy so that predictions with high entropy can be seen as uncertain, therefore discarded.

#### *3.3.2. Cosine similarity*

On the other hand, INFO slot predicts overall topic of the utterance and the whole parts may worth attention, making entropy an inappropriate choice. We thus threshold on the cosine similarity measure between the prediction and the closest word in ontology instead of entropy, so that the predictions that are not close enough to any subject are discarded.

Fig. 4 exhibits how the correct/wrong/absent values differ by the criteria above. As can be seen in (a) and (b), the entropy of correct values in regular slots are clearly distinguishable from the entropy of wrong or absent values. While (c) shows that the INFO slot can not be treated in the same way, (d) shows that the cosine similarity can be used instead for INFO slot.

We also tried to construct a neural network that gives the decision, namely the slot activation network, but it turned out to be perform poorly than two criteria above in practice. It will be dealt in further research.

## 4. RESULTS AND DISCUSSION

Table 3 summarizes the DSTC 5 final result. There are two kinds of evaluation method: Schedule 1 scores prediction at every utterance (utterance level evaluation) and Schedule 2 only scores prediction at the last utterance of every segment (segment level evaluation). Baseline tracker is based on fuzzy matching algorithm.

Among 9 teams participated the main task, We are team 6, and denoted as ours in Table 3. Our best entry accuracy scored 0.0491 in schedule 1 and 0.0643 in schedule 2, taking third place among all trackers submitted.

Although we are behind the performance of the tracker of team 2&4, we place more emphasis on the fact that our tracker is fairly simple and its performance is not heavily affected by the size of ontology unlike neural trackers in previous challenges. Sophisticated pre- and post-processing are not included as well, implying that the tracker proposed can be directly applied to other domains.

There still remains some room for improvement in our model; in training and predicting with our tracker, we treated the segments as independent instance disregarding the context among the segments from the same dialog. Since the information of dialog states appears a prior to current segment quite often, such context is crucial in determining dialog states. Adopting hierarchical structures such as HRED introduced in [15] would have helped for further improvements and remained as future work.

## 5. CONCLUSION

This paper presented our proposed tracker for DSTC 5. The proposed tracker is bi-directional LSTM with noble attention mechanism that can capture key phrases from utterances, enabling good performance in challenging task where the ontology is much larger than training vocabulary without sophisticated pre- and post-processing. Our concise model proved its efficiency by taking third place in the challenge. The code can be downloaded from our repository.[5]

[4] see https://github.com/seokhwankim/dstc5/tree/master/results for detailed results

[5] https://github.com/jys5609/DSTC5

## 6. REFERENCES


[1] Seokhwan Kim, Luis Fernando D'Haro, Rafael E. Banchs, Jason Williams, Matthew Henderson, and Koichiro Yoshino, "The Fifth Dialog State Tracking Challenge," in *Proceedings of the 2016 IEEE Workshop on Spoken Language Technology (SLT)*, 2016.

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

[3] Matthew Henderson, Blaise Thomson, and Steve Young, "Word-based dialog state tracking with recurrent neural networks," in *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, 2014, pp. 292–299.

[4] Matthew Henderson, Blaise Thomson, and Steve Young, "Robust dialog state tracking using delexicalised recurrent neural networks and unsupervised adaptation," in *Spoken Language Technology Workshop (SLT), 2014 IEEE*. IEEE, 2014, pp. 360–365.

[5] Koichiro Yoshino, Takuya Hiraoka, Graham Neubig, and Satoshi Nakamura, "Dialog state tracking using long short-term memory neural networks," in *Proceedings of the 7th International Workshop on Spoken Dialogue Systems (IWSDS)*, 2016.

[6] Alex Graves and Jrgen Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures," *Neural Networks*, pp. 5–6, 2005.

[7] Hasim Sak, Andrew W. Senior, and Françoise Beaufays, "Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition," *CoRR*, vol. abs/1402.1128, 2014.

[8] Laurens van der Maaten and Geoffrey E. Hinton, "Visualizing high-dimensional data using t-sne," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.

[9] Hongjie Shi, Takashi Ushio, Mitsuru Endo, Katsuyoshi Yamagami, and Noriaki Horii, "Convolutional neural networks for multi-topic dialog state tracking," in *Proceedings of the 7th International Workshop on Spoken Dialogue Systems (IWSDS)*, 2016.

[10] Nikola Mrksic, Diarmuid Ó Séaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve J. Young, "Neural belief tracker: Data-driven dialogue state tracking," *CoRR*, vol. abs/1606.03777, 2016.

[11] Alexander M. Rush, Sumit Chopra, and Jason Weston, "A neural attention model for abstractive sentence summarization," *CoRR*, vol. abs/1509.00685, 2015.

[12] Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomás Kociský, and Phil Blunsom, "Reasoning about entailment with neural attention," *CoRR*, vol. abs/1509.06664, 2015.

[13] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio, "Neural machine translation by jointly learning to align and translate," *CoRR*, vol. abs/1409.0473, 2014.

[14] Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O. K. Li, "Incorporating copying mechanism in sequence-to-sequence learning," *CoRR*, vol. abs/1603.06393, 2016.

[15] Alessandro Sordoni, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob Grue Simonsen, and Jian-Yun Nie, "A hierarchical recurrent encoder-decoder for generative context-aware query suggestion," in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. ACM, 2015, pp. 553–562.