Cross-Language Neural Dialog State Tracker for Large Ontologies Using Hierarchical Attention

Youngsoo Jang, *Student Member, IEEE*, Jiyeon Ham, Byung-Jun Lee, and Kee-Eung Kim, *Member, IEEE*

***Abstract*—Dialog state tracking, which refers to identifying the user intent from utterances, is one of the most important tasks in dialog management. In this paper, we present our dialog state tracker developed for the fifth dialog state tracking challenge, which focused on cross-language adaptation using a very scarce machine-translated training data when compared to the size of the ontology. Our dialog state tracker is based on the bi-directional long short-term memory network with a hierarchical attention mechanism in order to spot important words in user utterances. The user intent is predicted by finding the closest keyword in the ontology to the attention-weighted word vector. With the suggested methodology, our tracker can overcome various difficulties due to the scarce training data that existing machine learning-based trackers had, such as predicting user intents they have not seen before. We show that our tracker outperforms other trackers submitted to the challenge with respect to most of the performance measures.**

***Index Terms*—Dialog state tracking, attention mechanism, hierarchical attention mechanism, long short term memory, cross language.**

I. Introduction

A DIALOG state tracker identifies the user intent behind utterances, which is one of the most important tasks for dialog systems. Accurate dialog state tracking is essential for successful dialog since the system response is determined based on the tracked dialog state.

This paper is concerned with the dialog state tracker developed for the Fifth Dialog State Tracking Challenge (DSTC5). The task of DSTC5, which was to predict user intent within the tourist information domain, had the three major characteristics.

- *Scarce data:* The training data were very scarce compared to the range of topics and entities in the ontology database.
- *Human-to-human dialogs:* In the human-to-human dialogs, the range of topics is usually wide and the purpose of speech is often open-ended. Therefore, the training data is relatively scarcer, when compared to the human-to-system dialogs where dialogs are limited to specific topics and goal-oriented.
- *Cross-language:* The task was cross-language, where the tracker is developed for the target language using the training data in the source language along with their machine translation results.

Although we mainly report the performance results on DSTC5, we also provide results on the immediately preceding challenge, DSTC4. The task in DSTC4 was almost identical to that of DSTC5 except that it wasn't cross-language.

In this paper, we propose a slot-filling dialog state tracker that can effectively predict both in-vocabulary (IV) and out-of-vocabulary (OOV) values using a hierarchical attention mechanism. Our proposed model consists of two main components that predict user intent in different ways: one chooses from the value candidates of the slot defined by the ontology, while the other chooses from the words of the user utterance. The first component predicts IV values through attention-weighted output using attention to the slot-values which are defined by the ontology. The other component predicts OOV values through attention-weighted output using attention to the given utterance words, similar to zero-shot learning scenario [1], [2]. We use a hierarchical attention mechanism to adjust the weight of each components output according to the situation. Ideally, the model should learn which component's output is more reliable through the hierarchical attention mechanism. Our model is designed to work for both single-language and cross-language dialogs. Moreover, even when the output of our model is not exactly correct, the prediction will still be acceptable in terms of semantics since our model predicts a semantic word vector and chooses the closest keyword in the ontology.

Previous approaches using rule bases and neural networks make predictions by one-hot-encoding of slot values [3]–[6], which has a few limitations. Without natural language understanding components, they cannot predict the correct answer for OOV values. This is because the model parameters related to OOV values are not updated. In the more realistic dialogs we consider, OOV values are more abundant due to the scarcity of data compared to the size of the ontology. Correct prediction for OOV values is a challenge that must be addressed for realistic human-to-human dialog state tracking. Furthermore, most of the previous approaches tend to make poor predictions with long user utterances, because they try to encode the information

Manuscript received November 6, 2017; revised April 18, 2018 and June 18, 2018; accepted June 19, 2018. Date of publication July 2, 2018; date of current version August 8, 2018. This work was supported by the Ministry of Trade, Industry and Energy (MOTIE, Korea) under Industrial Technology Innovation Program (10063424, Development of distant speech recognition and multitask dialog processing technologies for in-door conversational robots). The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Eric Fosler-Lussier. *(Corresponding author: Youngsoo Jang.)*

The authors are with the Computer Science Department, Korea Advanced Institute of Science and Technology, Daejeon 34141, South Korea (e-mail: ysjang@ai.kaist.ac.kr; jyham@ai.kaist.ac.kr; bjlee@ai.kaist.ac.kr; kekim@cs.kaist.ac.kr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASLP.2018.2852492

TABLE I
EXAMPLE OF A SEGMENT FROM THE DSTC4 AND 5 DATASET; ONE SHOULD PREDICT THE SLOT-VALUE PAIRS IN **LABEL** GIVEN **TOPIC** AND **UTTERANCE**

Speaker	Utterance	Dialog state label
Guide	Yah so, you are uh going to do your proposal in the Botanical Gardens. (呀，你那个是去做你的建议在植物园。)	Topic: ATTRACTION INFO: Preference PLACE: Singapore Botanic Gardens ACTIVITY: Marriage proposal
Tourist	Well, actually no, I was thinking more uh I guess I think like I'm going to have to see the place first. (那倒不是，我是要多想那个我想我想我要去看的地方。)	
Tourist	But most probably. Yah, I guess. (但最可能的呀，我猜。)	
Tourist	Uh looking at the pictures, I guess I would choose the Botanical Gardens. (嗯，我想我会选择植物园看照片。)	
Guide	Alright. (还可以吧。)	

INFO, PLACE, and ACTIVITY are examples of slots whereas Preference, Singapore Botanic Gardens, and Marriage proposal are the corresponding values for such slots. All of the data in DSTC4 are provided in English only. In DSTC5, the training data is given in English but the validation and test data are given in Chinese, which is the original language of the data. The machine-translated results for each utterance are also given in DSTC5.

of every word in the utterance. Since this data includes many unimportant words, our model uses the attention mechanism to focus on a few important words for more effective predictions.

This paper is organized as follows: Section II provides descriptions of the DSTC4 and 5 datasets, as well as their main tasks. It also provides a brief description of the related works. Section III explains the various parts of our model, and Section IV discusses the detailed results of our model as they relate to DSTC4 and 5 tasks. Finally, Section V concludes this paper.

II. BACKGROUND

A. Dataset Description

DSTC4 and 5 used TourSG corpus to evaluate dialog state trackers. The TourSG corpus consists of dialog sessions for touristic information about Singapore collected from Skype calls between a tour guide and a tourist. The dialog states are labeled for each sub-dialog, which is denoted as a segment. A full dialog session is divided into segments based on their topical coherence. TABLE I presents an example of a segment in the DSTC4 and 5 datasets. Each segment consists of several utterances and all of the utterances in the same segment have the same dialog state.

The main task of DSTC4 and 5 is to predict the dialog state given the topic and the utterance. Each segment has a dialog state that consists of slot-value pairs. There can be one or more slot-value pairs for each segment, where each slot can be filled with multiple values. Each topic has a different set of slots. TABLE II summarizes the set of slots for each topic. The slots are categorized into two types: regular slots and the INFO slot. Regular slots are filled if specific values are directly discussed in the segment, and the INFO slot is filled with the corresponding subject when the specific values are not discussed. The ontology of valid values for each slot is also provided, of which the statistics are summarized in TABLE III. We also highlight that OOV rates are very high, training data covering only a fraction of the vocabulary. DSTC5 differs from DSTC4 in that the amount of data is larger, and the task is cross-language. TABLE IV

TABLE II
LIST OF EXISTING TOPICS AND CORRESPONDING SLOTS

Topic	Set of slots
Accommodation	INFO, PLACE, TYPE_OF_PLACE, NEIGHBOURHOOD
Attraction	INFO, PLACE, TYPE_OF_PLACE, NEIGHBOURHOOD, ACTIVITY, TIME
Food	INFO, PLACE, TYPE_OF_PLACE, NEIGHBOURHOOD, CUISINE, DISH, DRINK, MEAL_TIME
Shopping	INFO, PLACE, TYPE_OF_PLACE, NEIGHBOURHOOD, TIME
Transportation	INFO, TYPE, TO, FROM, LINE, STATION, TICKET

There are a total of 30 topic-slot pairs and there are value lists for each topic-slot pair in the ontology. Even though slot names are shared across topics, we considered them independently for training.

TABLE III
THE NUMBER OF VALUES FOR EACH SLOT IN THE ONTOLOGY

Slot	# values	# IV values	# OOV values
FROM	1197	99	1098
TO	1197	127	1070
PLACE	907	110	797
NEIGHBOURHOOD	209	29	180
DISH	193	69	124
STATION	95	18	77
CUISINE	67	26	41
ACTIVITY	62	57	5
TYPE_OF_PLACE	41	37	4
INFO	31	31	0
DRINK	19	8	11
TYPE	10	9	1
TIME	7	6	1
LINE	4	4	0
MEAL_TIME	4	4	0
TICKET	3	3	0

The number of IV and OOV values represent the amount of values, which are the slot-values included in the training data or not.

TABLE IV
SIZE OF DSTC4 AND 5 DATASET

Task	Set	Language	# dialogs	# utterances
DSTC4	Train	English	14	12,759
	Dev	English	6	4,812
	Test	English	9	7,848
DSTC5	Train	English	35	31,304
	Dev	Chinese	2	3,130
	Test	Chinese	10	14,878

For DSTC5, machine-translated Chinese data exist for Train set and machine-translated English for Dev/test data.

summarizes the statistics of data used in DSTC4 and 5. In DSTC5, the data is increased by 145% compared to DSTC4. Even with the increased amount of data, it is very small compared to the size of the vocabulary. All the data in DSTC4 are provided in English. In DSTC5, the training data is provided in English while the validation and test data are provided in Chinese, which is the original language used in data collection. The machine-translated result for each utterance is also given in DSTC5 in the form of top 5 probable translation for all dialogs. We use the original language data and top 1 result from the machine translation in our experiments.

B. Related Works

1) DSTC2 and 3: DSTC2 and 3 were the human-to-system dialog domains, and the goal of the task was to track the dialog state by using the results of automatic speech recognition and spoken language understanding (SLU) as inputs. Various methods such as rule bases, statistical models, and neural networks have been proposed in DSTC2 and 3 [7]–[10]. However, most of them rely on the SLU results. In this case, the performance of the dialog state tracking is greatly influenced by the performance of the SLU. In contrast, our proposed model directly tracks the dialog state from the utterance without the SLU results, and thus it can be trained end-to-end. We remark that [10] address the issues with OOV, but the work required the SLU results and significant feature engineering effort. In contrast, our proposed model can effectively handle OOV without feature engineering or SLU results.

2) DSTC4 and 5: The neural network has widely used in natural language processing (NLP) tasks, such as sentence classification, machine translation, and dialog systems [11]–[15]. In DSTC4, two teams proposed a neural network-based model. [16] reduce the task to the multi-domain text classification problem by focusing on filling the INFO slot. Using convolutional neural networks (CNN), they combine topic-shared and topic-specific structures. [17] suggest a tracker that integrates the baseline tracker and the LSTM to convey the information of previous utterances. Since the number of data is very small in DSTC4, neural network models resulted in a relatively poor performance compared to rule bases.

In DSTC5, three teams proposed neural network-based models. [4] propose a model that can solve the cross-language problem by extending the CNN sentence classification model [18] to multi-channel CNN for each language. Their model makes predictions by outputting one-hot-encoding of slot-values, which is not easily trainable for data with a much larger ontology than the training data. OOV values will not be correctly predicted with such approach without language understanding components. [5] suggest a tracker that integrates rule bases and LSTM attention model with bag-of-words encoding. In our previous work [19], we proposed a model that predicts a convex combination vector of words given to an utterance as an attention mechanism using a bi-directional LSTM. However, [5] and [19] did not solve the cross-language problem. In DSTC5, neural network approaches had better performance than rule bases, presumably due to the increased amount of data and complexity of the cross-language task.

In work apart from submissions to the DSTC challenges, CopyNet [20] is an interesting variant of the attention-based RNN encoder-decoder model that adopts the copying mechanism to deal with large ontologies. CopyNet has the ability to handle OOV words as proper nouns by copying consecutive sequences. For our task, the target values are not long enough to be predicted sequentially. We overcame this proper noun issue by improving the embedding model. Another approach uses external memory, such as a neural turing machine [21] and memory network [22]. The external memory can be used to reflect the previous prediction in the current prediction or to refer to the history of predictions. This enables predictions that reflect the context. However, since the prediction accuracy of each segment is low in the DSTC4 and 5 tasks, using the results to predict the next step may result in an accumulation of error. In this paper, we do not use external memory and focus on troubleshooting the cross-language issues instead.

We take a slightly different approach in order to resolve the problems that occur when we directly apply previous works to the challenging domains such as DSTC5. This paper aims to solve the problem of large ontologies and OOV values by outputting word vectors through attention mechanism. The basic structure is based on the work previously proposed in [19], but this paper solves the cross-language problem and OOV value predictions through a hierarchical attention mechanism. In the existing hierarchical attention network [23], two kinds of attention are used for words and sentences: attention to words in sentences and attention to sentences in dialogs. However, this paper summarizes utterances in two ways: through given words and through slot-values. Therefore, we pay hierarchical attention to the structures of the model, rather than only a given utterance.

III. MODEL

The DSTC task requires the prediction of slot-value pairs for each utterance. Typically this task is resolved by dividing it into two parts: a slot prediction and a value prediction. By selecting a slot first, we can reduce the scope of values considerably. However, with such a limited dataset, it is difficult to train accurate predictors, and inaccurate individual predictors can lead to a catastrophic failure when put together. Therefore, we take

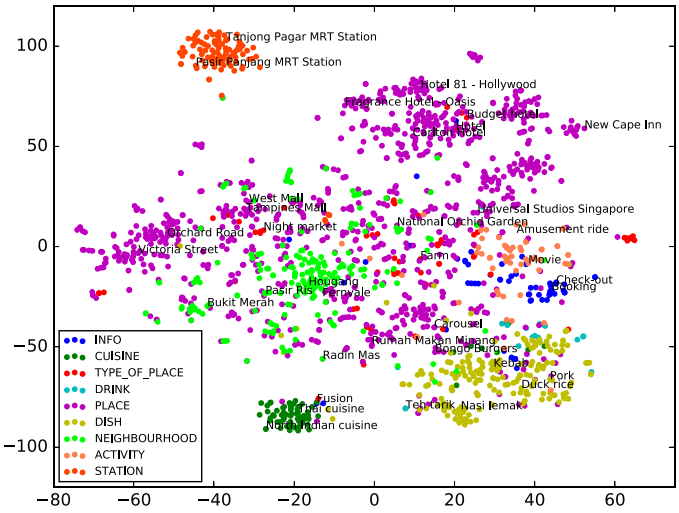


Fig. 1. Two-dimensional t-SNE embeddings of key phrases in the ontology learned by Word2Vec. The figure shows clear clusters of words of similar concepts.

a more holistic approach and directly solve the value prediction by omitting the slot prediction. We embed each word in a vector and directly predict the value vector.

A. Preprocessing Utterances

1) Word embedding: We project words into a high dimensional space and maintain the relationship between them using Word2Vec [24], which is well known as an effective method for word embedding. To deal with cross-language data of DSTC5 that involves both English and Chinese, we trained different Word2Vec models for the English words and Chinese characters. Since each Chinese character has meaning, we embedded each Chinese character in a unit. We converted each English word and Chinese character into 200-dimensional embedding vectors. The ontologies of DSTC4 and 5 mostly contained proper nouns specific to Singapore, so we additionally used web-crawled data to train Word2Vec models. For the English Word2Vec model, more than 13 million English sentences crawled from TripAdvisor[1] were used as a training data. For the Chinese Word2Vec model, more than 27 million Chinese sentences crawled from Wikipedia were used as a training data. Fig. 1 shows the results of slot-values, represented by t-SNE [25].

2) Ontology hint vector: The ontology and utterance contain valuable information, as well as its own meaning. We construct the ontology hint vector to encode this three pieces of information.

- *Type of speaker:* There are two types of a speaker in the human-to-human dialog. In the case of DSTC4 and 5, the guide and tourist make turns one by one. We encode speaker information for the 2-dimensional one-hot vector.

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

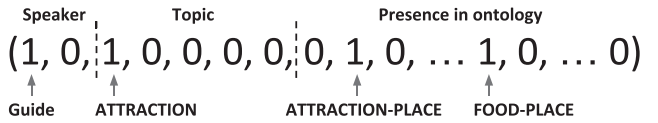


Fig. 2. The example of ontology hint vector for Gardens in the TABLE I. In this example, the speaker is Guide, the topic is ATTRACTION, and Gardens is in the PLACE slot for each topic in ontologies. Therefore, a vector with the corresponding values of 1 and the rest of 0 is the ontology hint vector of Gardens for the example.

- *Type of topic:* The topic information of each dialog is given. In the case of DSTC4 and 5, five types of topic exist. We encode topic information to 5-dimensional one-hot vector.
- *Presence in ontology:* Whether or not the word exists in the ontology of each topic-slot pair is important information. We construct 30-dimensional one-hot vector to encode this information (The number of topic-slot pairs is 30 in DSTC4 and 5).

We construct a 37-dimensional one-hot vector to encode this information, as illustrated in Fig. 2. We concatenate an ontology hint vector to a word-embedding vector and use it as an input for the value prediction network.

B. Model Architecture

We propose neural trackers that can be trained end-to-end. Fig. 6 represents the overall architecture of our tracker. Our proposed model predicts a word vector for a given utterance as an input and summarizes it as a vector. We use two different ways to generate a summary vector from a given utterance: the first is a convex combination based on slot-values and the second is a convex combination based on given words of utterance. The structure of our model is divided into two functional parts: a slot-value convex combination model and an utterance-words convex combination model, which produce outputs formed by taking the weighted sum of the embedding vectors of utterance words and slot-values. This section explains the attention-based keyword extraction model, which is the basic building block of our model, and then explains the core structures of our model that we built with the blocks.

1) Attention-based keyword extraction model: The segments assigned to DSTC4 and 5 have a maximum length of 510 words in English and 837 characters in Chinese. Since the direct encoding of such a long segment is itself a very challenging problem, we used the attention mechanism to choose the answer from the convex combination of candidates using bi-directional LSTM. We considered an utterance $\mathbf{u}$ that consists of N words $(u_1, u_2, \ldots, u_N)$. Each word is represented by a word-embedding vector $(\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_N)$ and an ontology hint vector $(\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_N)$. The input of the bi-directional LSTM is composed of the concatenation of the word-embedding vector and the ontology hint vector of each word:

$$\mathbf{X} = [\mathbf{w}_1 \oplus \mathbf{c}_1, \mathbf{w}_2 \oplus \mathbf{c}_2, \ldots, \mathbf{w}_N \oplus \mathbf{c}_N] \tag{1}$$

where $\oplus$ denotes the vector concatenation. Each word will then have the cell value of the bi-directional LSTM:

$$\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_N] \tag{2}$$

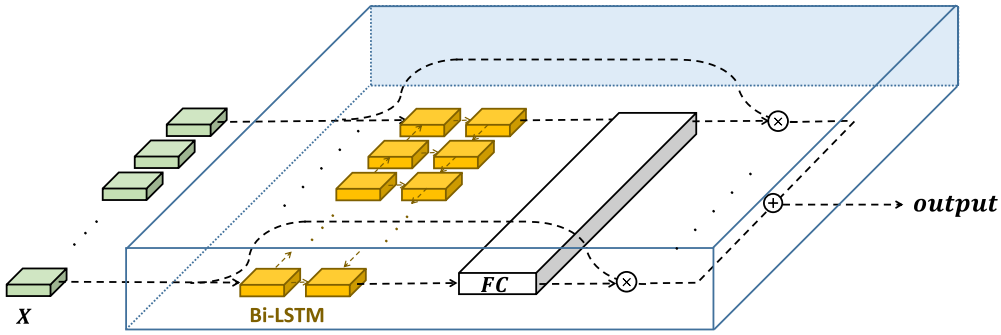


Fig. 3. Attention-based keyword extraction model (Model 1). This model uses $\mathbf{X}$, a vector composed of word vectors from utterance, as inputs, and it predicts a summary vector of given utterance through the attention mechanism using bi-directional LSTM.

The cell values are passed to the time-distributed dense networks (networks that share weights are equally applied to the cell values) for each slot to output a single scalar per word that represents the importance of the corresponding word. These scalars are normalized by applying the softmax function to the words in the utterance and will be denoted by attention weights to the words:

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

where $\mathbf{a}^s \in \mathbb{R}^N$ is the attention weight vector, $\mathbf{w}^s$ is the weight vector of the dense network, σ is a softmax function, and s denotes a specific slot. As an attention mechanism, we calculated the output vector by a weighted sum of word vectors with the weights from the 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 $\mathbf{a}^s$. The attention weight is calculated for all slots, and the model obtains a vector for all slots. It can now be said that this vector contains the essential information, which is a combination of important words for each utterance. In this model, the outputs are the convex combination of vectors of given words in user utterance. Since it is possible to extract a keyword from the context without understanding proper noun, this method is effective for predicting the OOV values, which the model has not seen while being trained. We construct the entire model based on the attention-based keyword extraction model proposed here. The attention-based keyword extraction model defined above will be called Model 1 in the following.

2) Slot-value convex combination model: The first part of our proposed model is the slot-value convex combination model. The structure of this model is presented in Fig. 4. We make a use of a list of possible values for each slot provided in the tasks of DSTC4 and 5. Let $\mathbf{Z}^s$ be the word vectors of the possible values for each slot: $\mathbf{Z}^s = [\mathbf{z}_1^s, \mathbf{z}_2^s, \ldots, \mathbf{z}_M^s]$ (M represents the number of possible values in slot s). Each $\mathbf{z}_i^s$ is a word embedding vector that represents the values in slot s through the Word2Vec model that we trained. However, in case of multi-word values, a normalized mean of their vectors was used. Our goal here is to get the weighted sum vector based on how probable the possible values are. DSTC5 includes data for two languages: English and Chinese. Each English utterance has $\mathbf{X}$, which is the list of word representation vectors. Likewise, each Chinese utterance has $\mathbf{Y}$, which is the list of character representation vectors. (Each word/character representation vector is made by concatenating the word/character embedding vector and ontology hint vector.) When $\mathbf{X}$ and $\mathbf{Y}$ pass through Model 1, it predicts a summary vector for English and Chinese as $\mathbf{x}$ and $\mathbf{y}$. (We do not share the weight of each Model 1, since English and Chinese lie in different embedding spaces.) We use a vector $\mathbf{k}$ that concatenates summary vectors for English and Chinese as a feature vector representing the corresponding utterance: $\mathbf{k} = \mathbf{x} \oplus \mathbf{y}$, where $\oplus$ denotes the vector concatenation. The feature vector is passed to the fully connected layers for each slot to formulate a single scalar per slot-values that represents the importance of the corresponding value. These scalars are normalized by applying softmax function and are further denoted as attention weights for the slot-values:

$$\mathbf{a}_v^s = \sigma(\mathbf{W}_1^s \mathbf{k}) \tag{5}$$

where $\mathbf{a}_v^s \in \mathbb{R}^M$ is the attention weight vector, $\mathbf{W}_1^s$ is the weight matrix of the dense network, σ is the softmax function, and s denotes a specific slot. The final output of this model, $\mathbf{v}_1^s$, is obtained through the weighted sum of attention $\mathbf{a}_v^s$ and slot-value vector $\mathbf{Z}^s$:

$$\mathbf{v}_1^s = \sum_{i=1}^{M} a_{v,i}^s \mathbf{z}_i^s \tag{6}$$

where $a_{v,i}^s$ is the scalar value at index i of $\mathbf{a}_v^s$, and $\mathbf{z}_i^s$ is the slot-value vector at index i of $\mathbf{Z}^s$. In this model, the outputs are the convex combination results for the slot-value vectors, and can effectively predict the IV values, which are the slot-values included in the training data. If the slot-value is not in the training data, however, a proper prediction cannot be made. Therefore, we propose an utterance-words convex combination model to complement the prediction in such cases. In the following, the slot-value convex combination model defined above will be called Model 2.

3) Utterance-words convex combination model: As we used the attention based convex combination method for slot-values above, the same method can be applied to words in the given utterance for summarization. We propose a model that is able to predict the dialog state correctly even if the model hasn't seen the slot-value from the training data. Fig. 5 shows the architecture of the utterance-words convex combination model.

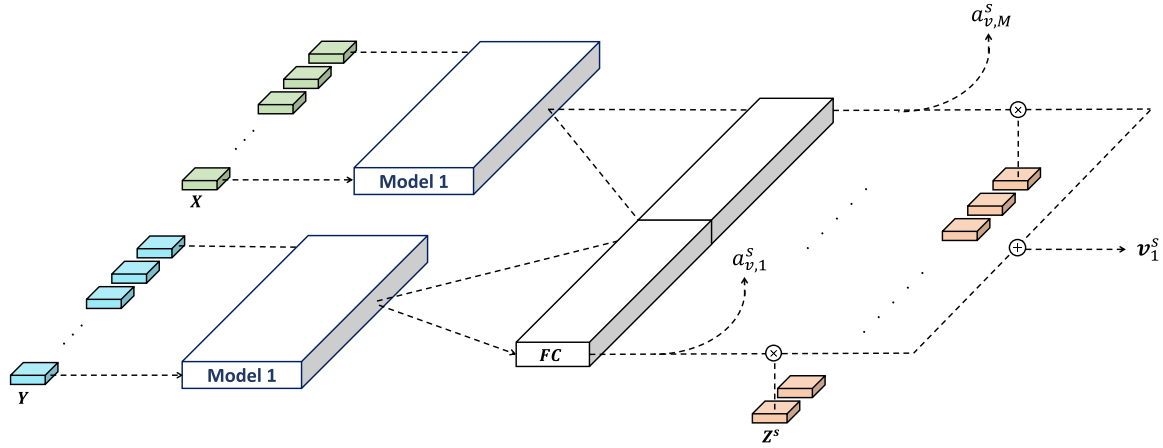


Fig. 4. Slot-value convex combination model (Model 2). This model predicts a summary vector that uses attention to slot-values $\mathbf{Z}^s$. It is also applicable to the cross-language domain, where $\mathbf{X}$ and $\mathbf{Y}$ represent different language inputs.

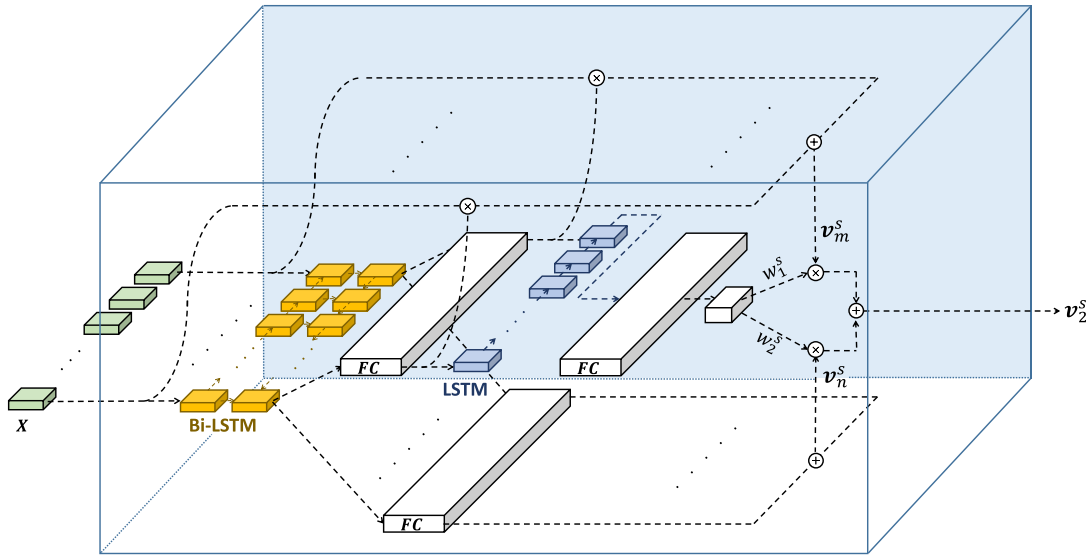


Fig. 5. Utterance-words convex combination model (Model 3). This model predicts a summary vector using attention to words in given utterance $\mathbf{X}$. The difference from Model 1 is that using a sentinel vector prevents the summary vector from outputting a meaningless summary when attention is unreliable.

The basic structure of this model is based on Model 1 (Fig. 3), but here we extend the model to be able to consider the case where the attention is not concentrated on any specific word. In the human-to-human dialog, there are cases where the topic or purpose is not clearly revealed. If there is no meaningful word in given utterance, it is not possible to obtain a meaningful vector by a simply convex combination of given words. Let $\mathbf{a}^s_w$, $\mathbf{v}^s_m$ be the attention weight and the output vector of Model 1 for each slot s. Model 1 unconditionally predicts the weighted sum of given word vectors regardless of the distribution of the attention weight, $\mathbf{a}^s_w$. However, in the worst case, if the model thought none of the words is important, the attention weights will be uniform over words and the results of Model 1 will be just an average of all of the word vectors. To prevent such a case, we adopted the sentinel vector method [23] so that a sentinel vector, which is an output of the additional structure, will be used instead of convex combination of input words if the attention is not reliable. (In Model 2, there is no such sentinel vector in order to catch only the slot-values that are revealed explicitly.) For the sentinel vector, we use a simple dense neural network that directly predicts a word vector:

$$\mathbf{v}^s_n = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{W}^s_2\mathbf{h}_i) \tag{7}$$

where $\mathbf{W}^s_2$ is the weight matrix of the dense network, $\mathbf{h}_i$ is the output of the bi-directional LSTM, N is the length of the utterance, and s denotes a specific slot. To determine whether the attention in Model 1 is reliable or not, an LSTM that receives the attention weight, $\mathbf{a}^s_w$, as an input, is used. It predicts the weights for the output of Model 1 and the sentinel vector. These weights are denoted as w^s_1 and w^s_2 for each slot, s, in Fig. 5. The final output of this model, $\mathbf{v}^s_2$, is obtained by the weighted sum of $\mathbf{v}^s_m$ and $\mathbf{v}^s_n$:

$$\mathbf{v}^s_2 = w^s_1\mathbf{v}^s_m + w^s_2\mathbf{v}^s_n \tag{8}$$

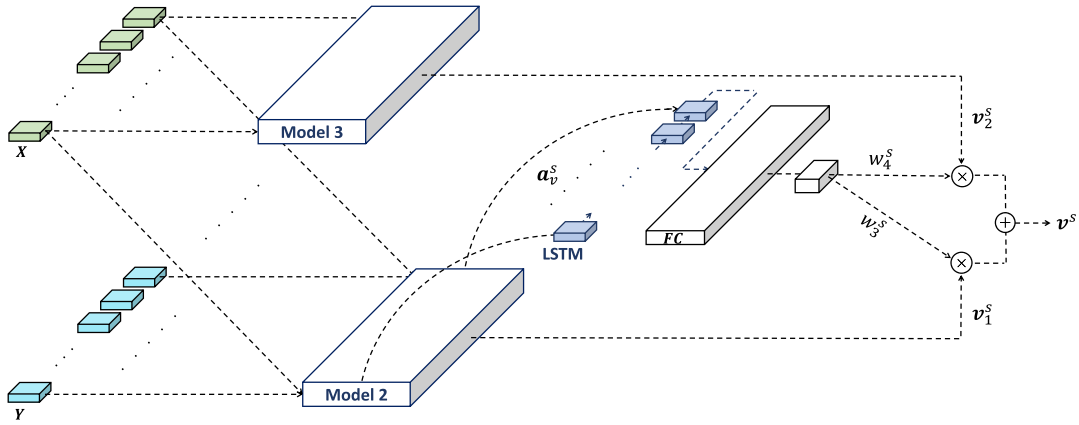


Fig. 6. Overall architecture (Model 4). The entire model proposed in this paper. This model can predict IV and OOV values effectively using Model 2 and 3.

This model is also effective for predicting OOV values, since the outputs are mainly the convex combination of vectors of given words in user utterance. The utterance-words convex combination model defined above will be called Model 3 in the following.

4) Overall architecture: Fig. 6 presents the overall architecture of our model, which combines Model 2 and 3. We first use Model 2 to predict the output vector based on the attention on the slot-values of the ontology. If the attention on the slot-values is not reliable in Model 2, it would be better to use Model 3 to make the prediction as we used the sentinel vector method above. An LSTM receives the attention weight of Model 2, $\mathbf{a}_v^s$, as an input, and it predicts the weight between the output of Model 2 and 3. For each slot, s, in Fig. 6, each weight is represented as w_3^s and w_4^s. The final output of this model, $\mathbf{v}^s$, is obtained by the weighted sum of $\mathbf{v}_1^s$ and $\mathbf{v}_2^s$:

$$\mathbf{v}^s = w_3^s \mathbf{v}_1^s + w_4^s \mathbf{v}_2^s \tag{9}$$

For every slot, s, we use $\mathbf{v}^s$ as the final output of the model and the model trains in the direction of maximizing the cosine similarity of $\mathbf{v}^s$ and the vector of true label value. By using hierarchical attention mechanism, our proposed model can effectively predict both IV and OOV values by choosing the appropriate model to predict. The overall architecture defined above is called Model 4 in the following sections.

C. Excluding Unreliable Predictions

The proposed model will give predictions for all slots while in usual dialog state tracking tasks there exists a choice of not making a prediction, which means that user intent is not clear enough. Therefore, we must decide whether to output a vector or not for each slot. We use both the entropy of attention weights and cosine similarity as the criteria to exclude unreliable predictions.

1) Entropy of attention weights: Our model determines the output through convex combination using attention to slot-values and attention to utterance words. Therefore, the output vector varies greatly depending on how these two attention weights are distributed. For example, if the attention distribution is highly peaked at a particular keyword, the result is more certain than being nearly uniformly distributed. We use entropy as a measure of the extent to which attention distribution is reliable. The entropy for attention, $\mathbf{a}^s$ is defined as $\mathcal{H}$, and we use the sum of entropy for each attention $\mathbf{a}_v^s$ and $\mathbf{a}_w^s$.

$$\mathcal{H} = -\sum_{i=1}^{N} a_i^s \log a_i^s \tag{10}$$

When the attention weight is concentrated on a specific word, we get low entropy. The entropy rises as the distribution of attention weight becomes more uniform.

2) Cosine similarity: Our model learns to maximize the cosine similarity between the output vector and true label vector. Therefore, it is possible to judge whether the output vector is reliable through the cosine similarity itself. For each slot, we calculate the maximum cosine similarity between the output vector and slot-values and use it to determine whether to exclude the output vector for that slot.

We use the validation data to determine the thresholds for the entropy and the cosine similarity for each slot. If the entropy is lower than its threshold while the cosine similarity is higher than its threshold, the result vector is predicted to the corresponding slot.

IV. Results and Discussion

There are two ways of evaluation in DSTC: Schedule 1 scores the prediction of every utterance (utterance-level evaluation) and Schedule 2 only scores the prediction of the last utterance of every segment (segment-level evaluation). Schedule 1 evaluates how quickly the dialog state is correctly tracked, and Schedule 2 evaluates the accuracy of the tracker's predictions when the information is sufficient. For each schedule, the algorithm is evaluated using four measures: accuracy, precision, recall, and the f-measure.

A. DSTC4 Results

Since the DSTC4 dataset consists of a single-language, we use the same structure except for the part where the Chinese input is used in Model 3. Using the DSTC4 dataset, we evaluate whether our algorithm works well even if training data is very scarce compared to the size of the ontology. TABLE V summarizes the overall results from the DSTC4 dataset (see [26] for more

TABLE V
THE SUMMARY OF RESULTS OF THE DSTC4 MAIN TASK IS SHOWN BELOW

Team	Schedule1				Schedule2			
	Accuracy	**Precision**	**Recall**	**F-measure**	**Accuracy**	**Precision**	**Recall**	**F-measure**
Baseline	0.0374	0.3589	0.1925	0.2506	0.0488	0.3750	0.2519	0.3014
Team3	**0.1212**	0.5393	**0.4980**	**0.5178**	**0.1500**	0.5569	**0.5808**	**0.5686**
Team4	0.1009	0.5583	0.3698	0.4449	0.1264	0.5666	0.4455	0.4988
Ours	0.0774	0.5088	0.3185	0.3918	0.0862	0.5402	0.3716	0.4403
Team1	0.0456	0.3876	0.3344	0.3591	0.0584	0.4384	0.3377	0.3815
Team2	0.0489	0.4440	0.2703	0.3361	0.0697	0.4634	0.3335	0.3878
Team6	0.0486	**0.5623**	0.2314	0.3279	0.0645	**0.5941**	0.2850	0.3852
Team5	0.0309	0.3039	0.2659	0.2836	0.0392	0.3398	0.2639	0.2971
Team7	0.0286	0.2768	0.1826	0.2200	0.0323	0.3054	0.2410	0.2694

We choose best trackers of each team and sorted based on the f-measure. The results of our proposed model (Model 4) is represented by **Ours**. Bold indicates the best score for each measure.

details). The baseline tracker uses a fuzzy matching algorithm. The following is the description for each team.

- Team 3: combined model of elaborate rule bases and random forests ([3])
- Team 4: combined model of rule bases and statistical method based on support vector machine (SVM) ([27])
- Team 1: topic-shared and topic-specific structures using CNN ([6])
- Team 2: probabilistic framework ([28])
- Team 6: probabilistic matching algorithm ([29])
- Team 5: no information about this team
- Team 7: combined model of rule bases and LSTM ([30])

TABLE V presents the results of our proposed model. Due to the nature of the DSTC4 dataset, which has a very small amount of data, our results show poor performance compared to rule bases, but the best performance among any the rest. Our accuracy is 0.0774 for Schedule 1 and 0.0862 for Schedule 2, which is 24% (0.0862/0.0697) better than the best approach except for rule bases. The f-measure is 0.3918 in Schedule 1 and 0.4403 in Schedule 2, which is 13% (0.4403/0.3878) better than the best approach except for rule bases.

B. DSTC5 Results

For the DSTC5 dataset, we evaluate whether our algorithm works well, even if the data is sparse and the ontology is large, as well as in the cross-language domain. TABLE VI summarizes the overall results for the DSTC5 dataset (see [30] for more details). The baseline tracker uses the same fuzzy matching algorithm as DSTC4. Baseline1 is the result of using English data, and Baseline2 is the result of using Chinese data. The following is the description for each team.

- Team 2: multi-channel CNN ([4])
- Team 4: combined model of rule bases and attention-based sequence-to-sequence learning ([5])
- Team 1: probability enhanced frame structure ([31])
- Team 6: attention mechanism by using bi-directional LSTM ([19])
- Team 5, 3: no information about these teams
- Team 8: combined model of author-topic model and SVM ([32])
- Team 9, 7: no information about these teams

TABLE VI presents the results of our proposed model. In most measures, our model has a better or similar score compared to the state-of-the-art algorithm (Team 2). The scores outperform those of Team 2, and more advantages to our model exist that were not represented numerically. Our model can be applied directly to the general domain. Its performance is not heavily affected by the size of the ontology and OOV values, unlike the neural trackers that make predictions by one-hot-encoding of slot-values. When learning through one-hot-encoding of slot-values using a neural network, the weight corresponding to the OOV value cannot be updated during the learning process. This problem cannot be avoided even if word embedding or other structures are different. Recently, zero-shot learning has been actively researched as a way to solve such problems [1], [2]. Zero-shot learning is to predict the values that have been omitted from the training data. The main idea of zero-shot learning is to generalize the unseen data with using knowledge and the relation between each label value. Our proposed model also yields generalization effects similar to zero-shot learning by convex combinations of given utterance words and slot-values.

C. Model Separate Results

TABLE VII summarizes the overall results of Model 1-4 in DSTC4 and 5. The results for each model identify the need for each part.

1) Effect of sentinel vector: Model 3 has better performance than Model 1 for all measures associated with both tasks in DSTC4 and 5. The difference between the two models is whether there the sentinel vector is included. In a convex combination with attention weight, the output of the worst-case scenario can be a simple average, meaning it contains meaningless information. The above results show that there are some cases where the attention is unreliably distributed in the DSTC4 and 5 test data. In this case, it can be confirmed that the sentinel vector is more effective.

2) Effect of each convex combination model: Our final model (Model 4) uses two kinds of convex combination methods. Model 2 uses the convex combination of slot-value vectors that is effective when the slot-value is included in the training

TABLE VI
THE SUMMARY OF RESULTS OF THE DSTC5 MAIN TASK IS SHOWN BELOW

Team	Schedule1				Schedule2			
	Accuracy	Precision	Recall	F-measure	Accuracy	Precision	Recall	F-measure
Baseline1	0.0250	0.1148	0.1102	0.1124	0.0321	0.1425	0.1500	0.1462
Baseline2	0.0161	0.1743	0.1279	0.1475	0.0222	0.1979	0.1774	0.1871
Ours	**0.0832**	**0.5273**	**0.3426**	**0.4153**	**0.0987**	0.5318	**0.3829**	0.4452
Team2	0.0788	0.5195	0.3315	0.4047	0.0956	**0.5643**	0.3769	**0.4519**
Team4	0.0583	0.4008	0.2776	0.3280	0.0765	0.4127	0.3284	0.3658
Team1	0.0417	0.3650	0.2795	0.3166	0.0612	0.3811	0.3548	0.3675
Team6	0.0491	0.4684	0.2193	0.2988	0.0643	0.4758	0.2623	0.3381
Team5	0.0330	0.3377	0.2318	0.2749	0.0520	0.3637	0.3044	0.3314
Team3	0.0351	0.3216	0.1515	0.2060	0.0505	0.3350	0.2045	0.2539
Team8	0.0192	0.3130	0.1048	0.1570	0.0214	0.3021	0.1046	0.1554
Team9	0.0231	0.1139	0.1090	0.1114	0.0314	0.1412	0.1487	0.1449
Team7	0.0092	0.4287	0.0431	0.0783	0.0107	0.4000	0.0441	0.0794

We choose best trackers of each team and sorted based on the f-measure. The result of our proposed model (Model 4) is represented by **Ours**. Bold indicates the best score for each measure.

TABLE VII
OVERALL RESULTS FROM MODEL 1–4 IN THE DSTC4 AND 5 DATASETS

Type	Model	Schedule1				Schedule2				Time
		Accuracy	Precision	Recall	F-measure	Accuracy	Precision	Recall	F-measure	(min)
DSTC4	Model1	0.0604	0.4467	0.2439	0.3155	0.0701	0.4636	0.3034	0.3667	36.7
	Model2	0.0564	0.4499	0.3103	0.3673	0.0663	0.4718	0.3537	0.4043	38.3
	Model3	0.0659	0.4870	0.2824	0.3575	0.0777	0.5086	0.3369	0.4053	53.3
	Model4	**0.0774**	**0.5088**	**0.3185**	**0.3918**	**0.0862**	**0.5402**	**0.3716**	**0.4403**	176.7
DSTC5	Model1	0.0533	0.4904	0.2318	0.3148	0.0696	0.4909	0.2690	0.3475	83.3
	Model2	0.0687	0.4804	0.2836	0.3567	0.0811	0.4778	0.3348	0.3937	218.3
	Model3	0.0757	**0.5591**	0.2803	0.3734	0.0888	**0.5773**	0.3157	0.4082	123.3
	Model4	**0.0832**	0.5273	**0.3426**	**0.4153**	**0.0987**	0.5318	**0.3829**	**0.4452**	560.0

For each model, we select the model with the lowest validation loss during 100 training epochs. We also report wall clock time taken to finish 100 training epochs. Bold indicates the best score for each measure.

TABLE VIII
RESULTS WITH AND WITHOUT ONTOLOGY HINT VECTOR (OHV)

Type	with/without OHV	Schedule1				Schedule2			
		Accuracy	Precision	Recall	F-measure	Accuracy	Precision	Recall	F-measure
DSTC4	without OHV	0.0544	0.4303	0.3093	0.3599	0.0748	0.4542	0.3441	0.3916
	with OHV	**0.0774**	**0.5088**	**0.3185**	**0.3918**	**0.0862**	**0.5402**	**0.3716**	**0.4403**
DSTC5	without OHV	0.0613	0.4778	0.3012	0.3695	0.0704	0.4908	0.3468	0.4064
	with OHV	**0.0832**	**0.5273**	**0.3426**	**0.4153**	**0.0987**	**0.5318**	**0.3829**	**0.4452**

Bold indicates the best score for each measure.

data. Model 3 uses the convex combination of word vectors in a given utterance to shorten the meaningful information if the slot-value is not included in the training data. We can confirm that the results of Model 4 are better than those of Model 2 and 3 for almost all measures. The roles of Model 2 and 3 are different, and both are essential elements that complement each other.

D. Effect of Ontology Hint Vector

TABLE VIII summarizes the results about the effect of ontology hint vector. We experimented on DSTC4 and 5 with all other conditions being the same except with or without the ontology hint vector. The experiment was conducted only for the Model 4 we proposed. We can confirm that the results of using the ontology hint vector are better for all measures.

E. Example of Attention Weight Visualization

As mentioned earlier, Model 2 and 3 have different roles. Model 2 makes effective predictions for the IV values, which is the value that exists in the training data. On the other hand, Model 3 makes an effective prediction for the OOV values, which is the value that does not exist in the training data. Fig. 7

Type	Attention weights										
Ontology value attention ($w_3^{INFO} = 0.894169$)	Activity	Architecture	Atmosphere	Audio guide	Booking	Dresscode	Duration	Exhibit	Facility	Fee	History
	Image	Itinerary	Location	Map	Name	Opening hour	Package	Place	Preference	Pricerange	Promotion
	Restriction	Safety	Schedule	Seat	Ticketing	Tour guide	Type	Video	Website		
Utterance word attention ($w_4^{INFO} = 0.105831$)	well	what	do	you	want	to	go	somewhere	else	er	it
	is	similar	to	for	example	where	i	can	go	shopping	where
	can	i	get	those	gourmet	food	really	ah	those	oh	to
	eat	food	with	the	shopping	heh	uh	huh	right		
True label	**INFO: Preference**										

Fig. 7. Example of attention weights on slot-values and utterance words for IV data. w_3^{INFO} and w_4^{INFO} represent the model attention weight for the INFO slot between Model 2 and 3. The darkness of color indicates attention.

Type	Attention weights									
Ontology value attention ($w_3^{DISH} = 0.295547$)	Satay	Hainanese chicken rice	Curry	Durian	Hokkien mee	Chilli crab	Tandoor	Carrot cake	Kaya toast	Bakkwa
	Fish head curry	Mee soto	Roti jala	Roti prata	Mooncake	Spring roll	Murukku	Mee siam	Naan	Biryani
	Bak chor mee	Ice cream	Vadi	Lor mee	Sugee cake	Rojak	Kuay chap	Tumpeng	Yusheng	Turtle soup
	Hum chim peng	Pork	Egg tart	Mee goreng	Zongzi	Fried pork	Roti john	Porridge	Curry puff	Laksa
Utterance word attention ($w_4^{DISH} = 0.704453$)	it	is	hot	in	front	this	is	called	laksa	then	this
	do	you	eat	spicy	food	huh	oh	yeah	i	have	done
	pretty	uh	very	addicting	because	i	have	some	food	oh	it
	is	like	this	this	is	just	the	right	flavour		
True label	**DISH: Laksa**										

Fig. 8. Example of attention weights on slot-values and utterance words for OOV data. w_3^{DISH} and w_4^{DISH} represent the model attention weight for the DISH slot between Model 2 and 3. The darkness of color indicates attention (The slot-value attention shows only the top 40 attention including the true label provided by the Laksa).

and Fig. 8 show how our model actually behaves, for each case described above. Fig. 7 provides an example of the prediction result for the IV values in the test data. That is, the training data contains the data about the Preference value. In this case, the weight for Model 2 is much higher than that of Model 3 ($0.894169 > 0.105831$), and the Preference has high attention in the slot-values. Fig. 8 shows an example of the prediction result for the OOV values in the test data. In other words, the training data does not include the data for the Laksa value. In this case, the weight for Model 3 is much higher than that of Model 2 ($0.704453 > 0.295547$), and the Laksa has high attention in the utterance words. Even though the Laksa is explicitly exposed to the utterance, Model 2 could not capture it. This is a common problem for previous approaches that output one-hot-encoding of slot-values. Through a hierarchical attention mechanism, we obtain advantages of both approaches that are outputting one-hot-encoding of slot-values and outputting word vectors. From this result, it can be confirmed that Model 2 and 3 are appropriately activated through a hierarchical attention mechanism with high weight, as appropriate for the situation.

Our model achieves 24% improved accuracy, compared to the best approach except for rule bases in the DSTC4 task. In the DSTC5 task, our model shows the best performance for almost all measures among all approaches. Moreover, our approach has three main contributions in addition to the numerical results. The first is that the output of our model has more information than other approaches because it outputs a vector that contains semantics. In fact, when ranking by cosine similarity between output vector and slot-values, more than 72% of true label value exists in the top 5 results of the cosine similarity. In other words, even if the answer is not correct, a vector with a similar meaning is outputted. The second is that prediction is possible even for OOV values, which is not included in the training data. Finally, since there is no pre-processing or handcrafted engineering, it can be easily applied to any domain. Our model is designed for both of single-language and cross-language dialogs. We can also train the model using data constructed in other languages.

This paper predicts the dialog state by considering only the utterance of the current segment. However, context is an important element in dialog state tracking. In particular, human-to-human dialogs that are dependent on previous dialogs are frequent. However, this paper focuses on issues related to large ontologies, cross-language, and OOV values; therefore, it does not consider context. Future research should construct a dialog state tracker that considers the context of dialogs by using external memory-based approaches [21], [22], [33] that can effectively control the dialog context.

V. Conclusions and Future Work

This paper proposes a cross-language dialog state tracker that works well even in situations where the amount of the training data is limited and the ontology is very large. Our proposed tracker is constructed using the neural network without any handcrafted engineering and can be applied to any domain without any pre-processing. By using bi-directional LSTMs and the attention mechanism, our model is effective even when the input is very long. In addition, each model is activated through a hier-

REFERENCES

[1] M. Norouzi *et al.*, "Zero-shot learning by convex combination of semantic embeddings," in *Int. Conf. Learning Representations*, 2014.

[2] M. Palatucci, D. Pomerleau, G. E. Hinton, and T. M. Mitchell, "Zero-shot learning with semantic output codes," in *Proc. Advances Neural Inf. Process. Syst.*, 2009, pp. 1410–1418.

[3] F. Dernoncourt, J. Y. Lee, T. H. Bui, and H. H. Bui, "Robust dialog state tracking for large ontologies," in *Dialogues With Social Robots*, New York, NY, USA: Springer, 2016.

[4] H. Shi *et al.*, "A multichannel convolutional neural network for cross-language dialog state tracking," *CoRR*, in *IEEE Workshop Spoken Language Tech.*, Dec. 2016.

[5] T. Hori *et al.*, "Dialog state tracking with attention-based sequence-to-sequence learning," in *Proc. IEEE Spoken Lang. Technol. Workshop*, 2016, pp. 552–558.

[6] H. Shi, T. Ushio, M. Endo, K. Yamagami, and N. Horii, "Convolutional neural networks for multi-topic dialog state tracking," in *Dialogues With Social Robots*, Singapore: Springer, 2017, pp. 451–463.

[7] K. Sun, L. Chen, S. Zhu, and K. Yu, "The SJTU system for dialog state tracking challenge 2," in *Proc. 15th Annu. Meeting Special Interest Group Discourse Dialogue*, 2014, pp. 318–326.

[8] J. D. Williams, "Web-style ranking and SLU combination for dialog state tracking," in *Proc. 15th Annu. Meeting Special Interest Group Discourse Dialogue*, 2014, pp. 282–291.

[9] S. Kim and R. E. Banchs, "Sequential labeling for tracking dynamic dialog states," in *Proc. 15th Annu. Meeting Special Interest Group Discourse Dialogue*, Philadelphia, PA, USA, Jun. 2014, pp. 332–336.

[10] M. Henderson, B. Thomson, and S. Young, "Word-based dialog state tracking with recurrent neural networks," in *Proc. 15th Annu. Meeting Special Interest Group Discourse Dialogue*, 2014, pp. 292–299.

[11] Y. Wang, M. Huang, X. Zhu, and L. Zhao, "Attention-based LSTM for aspect-level sentiment classification," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Austin, TX, USA, Nov. 1–4 2016, pp. 606–615.

[12] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Advances Neural Inf. Process. Syst.*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Red Hook, NY, USA: Curran Associates, Inc., 2014, pp. 3104–3112.

[13] K. S. Tai, R. Socher, and C. D. Manning, "Improved semantic representations from tree-structured long short-term memory networks," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics 7th Int. Joint Conf. Natural Language Proc.*, 2015, pp. 1556–1566.

[14] E. Song, F. K. Soong, and H. G. Kang, "Effective spectral and excitation modeling techniques for LSTM-RNN-based speech synthesis systems," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 25, no. 11, pp. 2152–2161, Nov. 2017.

[15] M. Sundermeyer, H. Ney, and R. Schlter, "From feedforward to recurrent LSTM neural networks for language modeling," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 23, no. 3, pp. 517–529, Mar. 2015.

[16] H. Shi, T. Ushio, M. Endo, K. Yamagami, and N. Horii, "Convolutional neural networks for multi-topic dialog state tracking," in *Proc. 7th Int. Workshop Spoken Dialogue Syst. (IWSDS)*, vol. 427, 2016, pp. 451–463.

[17] K. Yoshino, T. Hiraoka, G. Neubig, and S. Nakamura, "Dialog state tracking using long short-term memory neural networks," in *Proc. 7th Int. Workshop Spoken Dialogue Syst. (IWSDS)*, Saariselka, Finland, Jan. 2016, pp. 1–8.

[18] Y. Kim, "Convolutional neural networks for sentence classification," in *Proc.Conf. Empirical Methods Natural Language Proc.*, 2014, pp. 1746–1751.

[19] Y. Jang, J. Ham, B.-J. Lee, Y. Chang, and K.-E. Kim, "Neural dialog state tracker for large ontologies by attention mechanism," in *Proc. IEEE Spoken Lang. Technol. Workshop*, Dec. 2016, pp. 531–537.

[20] J. Gu, Z. Lu, H. Li, and V. O. K. Li, "Incorporating copying mechanism in sequence-to-sequence learning," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, 2016, pp. 1631–1640.

[21] A. Graves, G. Wayne, and I. Danihelka, "Neural turing machines," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, 2014, pp. 1631–1640.

[22] J. Weston, S. Chopra, and A. Bordes, "Memory networks," in *Proc. Int. Conf. Learn. Representations (ICLR)*, 2015, pp. 1–15.

[23] Z. Yang, D. Yang, C. Dyer, X. He, A. J. Smola, and E. H. Hovy, "Hierarchical attention networks for document classification." in *Proc. HLT-NAACL*, 2016, pp. 1480–1489.

[24] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *Proc. Int. Conf. Learn. Representations (ICLR)*, 2013, pp. 1–12.

[25] L. van der Maaten and G. E. Hinton, "Visualizing high-dimensional data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, 2008.

[26] S. Kim, L. F. D'Haro, R. E. Banchs, J. D. Williams, and M. Henderson, *The Fourth Dialog State Tracking Challenge*, Singapore: Springer, 2017, pp. 435–449.

[27] K. Sun, S. Zhu, L. Chen, S. Yao, X. Wu, and K. Yu, "Hybrid dialogue state tracking for real world human-to-human dialogues," in *Proc. INTERSPEECH*, 2016, pp. 2060–2064.

[28] M. Li and J. Wu, "The MSIIP system for dialog state tracking challenge 4," in *Dialogues With Social Robots*, Singapore: Springer, 2017, pp. 465–474.

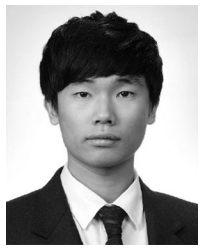
[29] J. Perez and W. Radford, "Probabilistic matching for dialog state tracking with limited training data," in *Proc. 7th Int. Workshop Spoken Dialog Syst. (IWSDS)*, 2016, pp. 1–6.

[30] S. Kim, L. F. D'Haro, R. E. Banchs, J. Williams, M. Henderson, and K. Yoshino, "The fifth dialog state tracking challenge," in *Proc. IEEE Spoken Lang. Technol. Workshop*, 2016, pp. 511–517.

[31] Y. Su, M. Li, and J. Wu, "The MSIIP system for dialog state tracking challenge 5," in *Proc. Spoken Lang. Technol. Workshop*, 2016, pp. 525–530.

[32] R. Dufour, M. Morchid, and T. Parcollet, "Tracking dialog states using an author-topic based representation," in *Proc. Spoken Lang. Technol. Workshop*, 2016, pp. 544–551.

[33] S. Sukhbaatar, A. Szlam, J. Weston, and R. Fergus, "End-to-end memory networks," in *Proc. Advances Neural Inf. Process. Syst. 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, Inc., 2015, pp. 2440–2448.



Youngsoo Jang received the B.S. degree in mathematical science and computer science, and the M.S. degree in computer science, in 2016 and 2018, respectively, from the Korea Advanced Institute of Science and Technology, Daejeon, South Korea, where he is currently working toward the Ph.D. degree in computer science. His current research interests include dialog system and deep learning.

Jiyeon Ham received the B.S. degree in physics and computer science, in 2016, from the Korea Advanced Institute of Science and Technology, Daejeon, South Korea, where she is currently working toward the M.S. degree in computer science.

Byung-Jun Lee received the B.S. and M.S. degrees in computer science, in 2013 and 2015, respectively, from the Korea Advanced Institute of Science and Technology, Daejeon, South Korea, where he is currently working toward the Ph.D. degree in computer science. His current research interests include Gaussian processes and Bayesian deep learning.

Kee-Eung Kim received the B.S. degree in computer science from the Korea Advanced Institute of Science and Technology, Daejeon, South Korea, in 1995, and the Sc.M. and Ph.D. degrees in computer science from Brown University, Providence, RI, USA, in 1998 and 2001, respectively. From 2001 to 2006, he was a Senior Software Engineer with Samsung SDS, South Korea, and a Senior Research staff member with Samsung advanced institute of technology, South Korea. In 2006, he joined the Faculty of Computer Science Department, KAIST. His research interests include representations and algorithms for sequential decision making problems in artificial intelligence and machine learning, including Markov decision processes, and reinforcement learning.