CDS: Contextual Spoken Language Understanding Model Based on Dual Semantics

Kun Ma1a*, Minghao Hu1, Fangzhao Li1 and Li Luo1
1School of Computer National University of Defense Technology, ChangSha, Hunan, 410073, China
3makun18@nudt.edu.cn  *Corresponding author’s e-mail: makun1995@qq.com

Abstract. Spoken language understanding (SLU) in multi-turn task-oriented dialogue systems usually needs to take historical contextual information into account. To achieve this, current approaches encode dialogue utterances with contextual information to improve the accuracy of slot filling and intent detection. However, this kind of approach ignores the different influences of contextual information on dialogue utterances. Some utterances with independent context will be misunderstood due to the addition of noisy contextual information. In this paper, we propose a Dual Semantics model(CSD) that is based on dual semantics. Each dialogue utterance is encoded based on independent semantics and contextual semantics, where the contributions of these two semantic information are dynamically decided. A large number of experiments on two datasets verify the effectiveness of our approach. Specifically, the accuracy of slot filling at the KVRET dataset increases by 3.3% compared to the baseline model. A dataset of psychological service appointments(PSA), is further collected for the verification of our approach. Compared with existing methods, the accuracy is improved by 4.9% on this dataset.

1. Introduction
Spoken language understanding (SLU) is one of the most important tasks in task-oriented dialogue systems[1-2]. This task usually contains two sub-tasks, namely slot filling and intent detection. Slot filling aims to find out semantic components in sentences (e.g., in the sentence "I would like to eat an apple.", the word "apple" should be marked as food). Intent detection aims to classify the overall intent of sentences. For example, given the sentence "I want to find a restaurant.", its intent should be identified as "looking for a restaurant". Since semantic components in sentences are highly related to their intent, slot filling and intent detection are now handled in the same model.[5-6]

In multi-turn spoken language understanding, contextual information is very important, because it helps the model understand the dialogue discourse with historical information. Take Figure 1 as an example, user gives "three" as a response for the question "How many tickets do you need?", if there is no such context information, it is impossible to determine whether it is the number of movie ticket or the number of days. Consequently, some studies try to improve the SLU performance by encoding the sentences with contextual information.[4][8]

The inputs of SLU are context $m = \{m_1, m_2, ..., m_n\}$ and the current utterance $x$ where the $n$ is the context length. The outputs of SLU are slots and intents of $x$ in conjunction with the context. the current mainstream models encode the $m = \{m_1, m_2, ..., m_n\}$ to contextual vector $h$, and combine the $h$ with the current utterance by the BiRNN model. Then they are used for the downstream prediction task.
However, there exists a main problems in these approaches: These approaches have not considered the context-free situation when encoding dialogue utterances. As shown in Table 1, The overall performance of the contextual model is better than that of the non-contextual model. But there are seven kinds of slots in which the performance of the contextual model is worst than that of the non-contextual model. So we infer that the contextual information brings noises to the model and some sentences don’t need the contextual information.

Table 1. The prediction results of seven types of slots on context-based and context-free model respectively for the KVRET dataset.

| Types of slots      | P          | Number of slots |
|---------------------|------------|----------------|
|                     | Context-based model | Context-free model |
| I_date              | 37.3       | 57.6           | 286           |
| I_distance          | 41.2       | 80.0           | 93            |
| I_event             | 75.5       | 78.9           | 169           |
| B_poi_type          | 74.6       | 75.1           | 180           |
| I_poi_type          | 39.2       | 59.6           | 121           |
| B_weather_attribute | 88.8       | 89.3           | 175           |
| I_weather_attribute | 72.2       | 92.9           | 19            |

Above all, the contributions are as follows:
1. A dual semantic mechanism is proposed to encode dialogue utterances so that the interference of contextual information on dialogue utterances can be effectively reduced.
2. Based on the above innovations, a contextual spoken language understanding model is proposed, which aims to jointly learn slot filling and intent detection.
3. A dataset of psychological service appointments, named PSA, is collected for the task of multi-turn spoken language understanding. A large number of experiments verify the validity of the model on the KVRET dataset and the PSA dataset.

2. The Method

2.1. Dual Semantic Mechanism
The model described above takes contextual information into account when encoding current user utterances. However, the language is complex, and the semantic dependence of each utterance on contextual information may be not consistent. For utterances that are less dependent or independent, this encoding pattern could potentially introduces ambiguity problems caused by noisy information, which eventually affects the performance of slot filling and intent recognition.
To address the above issue, we propose a dual semantic mechanism that combines both contextual semantics and independent semantics. Contextual semantics indicate that historical semantic information should be considered when calculating the semantics of current utterances, while independent semantics means that the encoding of current utterances only depends on the utterance's semantics alone. The whole computation process is defined as follows:

$$y = w_1(x, h) \cdot f(x, h) + w_2(x, h) \cdot g(x)$$  \hspace{1cm} (1)

In the above equation, x represents the current user utterance, h represents the contextual vector, f is the traditional contextual encoder, g is the independent semantic encoder, and w1(x, h) and w2(x, h) represent the weight of contextual semantics and independent semantics. w1(x, h) and w2(x, h) are the probability outputs of a sub neural network that takes x and h as inputs, and their sum is one.

The weights of w1(x, h) and w2(x, h) in Equation 1 have an important influence on the performance of dual semantics, so how to get a reasonable weight of w1(x, h) and w2(x, h) is a key problem.

2.2. Contextual and Independent Semantic Combination Weight

Semantic combination refers to how to decide w1(x, h) and w2(x, h) in Equation 1. Different situations have different degrees of context dependence, and [13] has verify this through a large number of experiments.

For method (A), we directly make the w1 and w2 as two learnable parameters. Since there are only two parameters, we can learn the different degrees of dependence of different task but we can’t learn the different degrees of dependence of different sentences in the same task. Hence, it is suitable to the SLU task that doesn’t have complex contextual information.

Typically, it is an accepted fact that the semantic complexity of each task background is different, so the semantic combination method(A) used above is only suitable for semantically simple task background. It is difficult to be competent in semantically complex tasks. To this end, a second semantic combination method is designed to calculate w1(x, h) and w2(x, h) for different utterance, which mainly consists of two steps. The first step(a) obtains the currently utterance vector by BiLSTM, and the second step(b) is dot product by utterance vector and contextual vector. w1(x, h) is processed by the sigmoid function, and then w2(x, h) is obtained by using the restriction of w1(x, h) + w2(x, h) = 1, and w1(x, h), w2(x, h) ≥ 0. This semantic combination method increases the number of parameters and enhances the learning ability of the model, but it is easy to overfitting when the data is insufficient or the semantics is simple. We refer this method as method(B).

2.3. Architecture of Contextual SLU

In this section, we will introduce the improved context of SLU architecture based on the above work, which is shown in Figure 2. The architecture mainly consists of five parts, namely encoder module, memory retrieval module, semantic combination module, semantic depth fusion module, prediction module. Below we will describe them in more details.
Figure 2. Architecture of Contextual SLU. It is worth noting that our improvement of this paper is in the dotted frame.

2.3.1. Memory Encoder
Memory Encoder encodes each historical utterances into a numerical vector. Let \( m = \{m_1, m_2, \ldots, m_{k-1}\} \) denote each historical utterances, where \( m_i \) represents the historical utterances \( i, 1 \leq i \leq k-1 \). It will be stored, \( k-1 \) represents the length of the historical utterances, and the vector representation of each historical utterances \( M = \{M_1, M_2, \ldots, M_{k-1}\} \) is obtained by Memory Encoder. Here, the BiLSTM is used as the Memory Encoder. The Equation is as follows:

\[
M = BiLSTM(m)
\] (2)

2.3.2. Sentence Encoder
The Sentence Encoder encodes the current user utterance into a numerical vector. Let \( x = \{x_1, \ldots, x_n\} \) denote the result of the current user utterance after embedding. Where \( x_i \) represents the corresponding word vector, \( n \) represents the length of the sentence, and \( x \) is also encoded by BiLSTM to get \( c = \{c_1, c_2, \ldots, c_n\} \), at the same time, the vector spliced together by the last vectors in the \( c_1 \) and \( c_n \) directions is recorded as \( C \), and the Equation is as follows:

\[
c = BiLSTM(x), C = [c_1; c_n]
\] (3)

2.3.3. Independent Semantic Encoder
Independent Semantic Encoder does not take into account the contextual information to encode the current user utterance, and the result is independent semantics. Because independent semantics have no contextual information, it will maintain their semantic information. Using the BiLSTM model, only 2.3.2 in \( c = \{c_1, c_2, \ldots, c_n\} \) as input, the independent semantics \( I = \{I_1, I_2, \ldots, I_n\} \) are obtained. The Equation is as follows:

\[
I = BiLSTM(c)
\] (4)

2.3.4. Contextual Semantic Encoder
Contextual Semantic Encoder, considering the contextual information to encode the current user utterance, the result is contextual semantics. Because contextual semantics introduces contextual information, it will contain historical information. Here we will use the [9] method, using BiLSTM encode, with (3) in \( c = \{c_1, c_2, \ldots, c_n\} \) as input, the contextual vector \( h \) in (6) is used as the initial hidden state of BiLSTM, and the contextual semantics \( E = \{E_1, E_2, \ldots, E_n\} \) are obtained. The Equation is as follows:

\[
E = BiLSTM(c, h)
\] (5)
2.3.5. Memory Retrieval
Memory Retrieval function is to retrieve the history calculation contextual vector $h$ according to the current user utterance. In this paper, we will use the method of [9] to get the historical representation vector $M = \{M_1, M_2, ..., M_{n-1}\}$ and (3) after the current user representation vector $C$ is obtained, $C$ and each item in $M$ are spliced to $Z = \{[M_1; C], [M_2; C], ..., [M_{n-1}; C]\}$ is then input as a BiGRU network, and the last vector in the front and back directions is the contextual vector $h$. The equation is as follows:

$$h = BiLSTM(Z), h = [tmp_1; tmp_2]$$  \hfill (6)

2.3.6. Semantic Combination
The dual semantics is obtained by combining independent semantics with contextual semantics. After obtaining independent semantics $I = \{I_1, I_2, ..., I_n\}$ and contextual semantics $E = \{E_1, E_2, ..., E_n\}$ by (4) and (5), respectively, using the method mentioned in 2.2 above as the input of semantic combination, the dual semantics $y = \{y_1, y_2, ..., y_n\}$, The Equation is as follows:

$$y = w_1(x, h) \ast E + w_2(x, h) \ast I$$  \hfill (7)

2.3.7. Prediction
After getting $y = \{y_1, y_2, ..., y_n\} \text{ at the same time, the vector spliced together by the last vectors in the } y_i \text{ and } y_n \text{ directions is recorded as } S, \text{ we use two fully connected networks, } FFO, FFS. FFO \text{ is used for slot filling and } FFS \text{ is used for intent recognition. The Equation is as follows:}$

$$Slot_i = FFO(y_i)$$  \hfill (8)

$$Intent = FFS(S)$$  \hfill (9)

2.3.8. Loss Function
In order to adjust the parameters of SLU, the loss function during training is defined as follows:

$$loss_1 = \log(p(\text{Intent} | x_1, ..., x_n))$$  \hfill (10)

$$loss_2 = \sum_{i=1}^{n} \log(p(Slot | x_1, ..., x_n))$$  \hfill (11)

$$Loss = loss_1 + loss_2$$  \hfill (12)

In the above Equation, $x$ represents the current user utterance, $n$ represents the length of the user utterance, $loss_1$ is the Loss function of intent recognition, $loss_2$ is the Loss function of Slot filling, and $LOSS$ is the Loss function of the whole model.

3. Experiments
In our experiments, two datasets, KVRET[7] and PSA, are adopted to verify the performance of our model. A large number of experiments are carried out and the results show that our method effectively improves the original model.

3.1. Dataset
KVRET is a multi-turn task-oriented dialogue dataset from vehicle assistants in English. It has been collected according to Wizard-of-Oz scheme[3]. KVRET consists of 3031 conversations in 3 different domains and 25 types of slots, 2681 in total. Each domain contains only one intent including calendar plans, weather queries, and navigation queries. The average number of conversations is 5.25 rounds.

| Datasets | train | test | Avg.turns |
|----------|-------|------|-----------|
| KVRET    | 2425  | 606  | 5.25      |
| PSA      | 596   | 150  | 14.51     |

To verify the model performance in a multi-lingual environment, we collected a dataset of psychological service appointments in Chinese, named PSA. PSA is also a multi-turn task-oriented
dialogue dataset and it contains 746 conversations and the average number of conversations reach 14.51 rounds. Two roles are involved in PSA: (1) Staff members in charge of psychological appointment service (T). (2) Students who want to make a psychological service appointment (S). Compared with KVRET, PSA contains more intentions, 11 intents from T and 10 intents from S. At the same time, there exist 10 slots in PSA and each slot is labeled as a unit in IOB format.

Each dataset is divided into two sets for model training and testing in our experiments. The parameters of the dataset are shown in Table 2.

3.2. Experimental Configuration

- **NoMem:** It represents such a SLU model without contextual information.
- **SDEN:** It is the baseline model which is proposed by Google[9]
- **MNM:** It uses attention mechanism to get the contextual vector $h$, and then combines the contextual vector $h$ with current utterance $C$. [10]
- **MNMDLJ:** It uses joint learning method. [12]
- **CDS(A):** model with the semantic combination mechanism based on method(A).
- **CDS(B):** model with the semantic combination mechanism based on method(B).
- **CDS(A+B):** model with the semantic combination mechanism based on method(A) and method(B).

In our experiments, our training batch size is 64 and all the models are trained with Adam[14] optimizer with default parameters. We conduct training up to 30 epochs. The word embedding size is 300. All the hidden sizes of BiRNN are 256. The learning rate is 10^{-4}. The dropout rate is set to be 0.3.

Table 3. The performance of CDS on the datasets, KVRET and PSA.

| Datasets | Models  | Slots | Intent |
|----------|---------|-------|--------|
|          |         | P     | R      | F1    |
| KVRET    | NoMem   | 68.5  | 66.5   | 66.0  | 84.5  |
|          | MNM     | 72.2  | 74.0   | 72.1  | 97.7  |
|          | SDEN    | 70.4  | 72.6   | 68.5  | 96.7  |
|          | MNMDLJ  | 76.3  | 79.3   | 76.4  | 98.3  |
|          | CDS(A)  | 79.6  | 79.5   | 78.1  | 99.1  |
|          | CDS(B)  | 72.4  | 80.5   | 73.8  | 98.2  |
|          | CDS(A+B)| 75.7  | 70.8   | 71.1  | 98.5  |
| PSA      | NoMem   | 67.1  | 57.4   | 59.0  | 70.8  |
|          | MNM     | 72.0  | 65.4   | 66.1  | 70.9  |
|          | SDEN    | 70.3  | 58.3   | 61.0  | 71.2  |
|          | MNMDLJ  | 75.6  | 67.9   | 70.3  | 72.2  |
|          | CDS(A)  | 74.7  | 69.7   | 70.6  | 72.0  |
|          | CDS(B)  | 78.5  | 68.3   | 71.0  | 72.1  |
|          | CDS(A+B)| **80.5** | **69.7** | **72.6** | **72.5** |

3.3. Performance of CDS

A group of experiments are conducted to verify the performance of our model on the two datasets, KVRET and PSA. The results are shown in Table 3.

Compared to the baseline model, the slot P score of CDS is increased by 3.3% and the slot F1 score of CDS is higher by 1.7% on the dataset KVRET. Our model increased by 4.9% in the slot P score and 2.3% in the slot F1 score on the dataset PSA. These results show that our model has greatly improved the accuracy of slot filling when the accuracy of intent detection is remained.

According to the results of experiments, the performance of Contextual model is better than NoMem on both the datasets, KVRET and PSA. It indicates that contextual information is useful for spoken language understanding.

Then, we adopt both the methods, method(A) and method(B), to combine the independent semantics and contextual semantics on KVRET and PSA respectively. Then compare the performance
of CDS with other models under the same conditions. All the models with the semantic combination mechanism are significantly better than the baseline model. It shows that dual semantics can improve the accuracy of slot filling effectively.

It's worth noting that method(A) is slightly better than the method(B) and the method(A+B) on the dataset KVRET. And on the dataset PSA, the Method(A+B) has the best results. Since KVRET semantics are relatively simple than PSA, the simpler method(A) can produce better results in KVRET. Similarly, the complicated method(A+B) is more suitable for PSA.

According to our experiments, the model with method(A) can increase the slot P score by 3.3% and the slot F1 score by 1.7% on the dataset KVRET. And the model with method(A+B) can increase the slot P score by 4.9% and the slot F1 score by 2.3% on the dataset PSA.

4. Related work
The traditional dialogue systems usually adopt the pipeline[1-2] model. It regards the two tasks of spoken language understanding, slot filling and intent detection, as two independent steps. However, this way can lead to error transmission and affect the performance of the dialogue system. Therefore, the current research mainly focuses on the joint training of the two tasks [5-8]. It trains the models based on the correlation of slot filling and intent detection and significantly improves the performance of spoken language understanding.

Both the traditional pipeline model and the current joint training model need encode the dialogue utterances firstly. In multi-turn spoken language understanding, contextual information from historical sentences is helpful to understand the current sentence. Some research work tries to improve the effect of slot filling and intent detection by fusing contextual information into the model. Their main idea is to calculate a contextual vector which is added in as initial values of the model. There are three ways to improve the model in related works.

1. Change the calculate method of the contextual vector.[10] calculates the contextual vector by the attention mechanism, and get the improvement of the accuracy.[9] uses the recurrent neural network to calculate the contextual vector and solves the problem of the variability of contextual information length. It also get the improvement of the accuracy.
2. Enrich the model’s feature information. [11] introduce more feature information, like role information, time information and so on.
3. Use the joint training method to further improve the accuracy of the model. Since the logic of dialogue is also the important information, [12] introduces the joint learning method with dialogue logistic inference.

Though these approaches improve the effect of contextual spoken language understanding, they only focus on the addition of contextual information to the dialogue utterance. The meaning of the sentence itself has not been paid enough attention. It leads to misunderstanding of some utterances due to the interference of contextual information.

5. Conclusions
Contextual information usually helpful to understand the conversation correctly. Therefore, multi-turn task-oriented dialogue systems usually add historical contextual information to the current dialogue utterance, which can improve the accuracy of slot filling and intent prediction efficiently. However, contextual information has different influence on dialogue utterance. Some utterances may be misunderstood due to the addition of noisy contextual information. To solve these problems, a Dual Semantics model(CDS) is proposed based on dual semantics. A large number of experiments on two datasets are carried out to verify the effectiveness of our model. Compared with existing methods, the accuracy of slot filling and intent detection improves significantly on both datasets.

References
[1] Chen, H., Liu, X., Yin, D., & Tang, J.. (2017). A survey on dialogue systems. Acm Sigkdd Explorations Newsletter, 19(2), 25-35.
[2] Chen, Yunnung, Asli Celikyilmaz, and Dilek Hakkanitur. (2018). Deep Learning for Dialogue Systems. International conference on computational linguistics. 25-31.

[3] Wen, Tsunghsien, et al. (2017). A Network-based End-to-End Trainable Task-oriented Dialogue System. Conference of the association for computational linguistics. 438-449.

[4] Bhargava, Aditya, et al. (2013). Easy contextual intent prediction and slot detection. International conference on acoustics, speech, and signal processing 8337-8341.

[5] Liu, Bing & Lane, Ian. (2016). Attention-Based Recurrent Neural Network Models for Joint Intent Detection and Slot Filling. 685-689. 10.21437/Interspeech.2016-1352.

[6] Zhang, Chenwei, et al. (2019). Joint Slot Filling and Intent Detection via Capsule Neural Networks. Conference of the association for computational linguistics. 5259-5267.

[7] Eric, Mihail, et al. (2017). Key-Value Retrieval Networks for Task-Oriented Dialogue. Annual meeting of the special interest group on discourse and dialogue. 37-49.

[8] Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. (2015). End-to-end memory networks, in Advances in neural information processing systems, pp. 2440–2448.

[9] Bapna, Ankur, et al. (2017). Sequential Dialogue Context Modeling for Spoken Language Understanding. Annual meeting of the special interest group on discourse and dialogue. 103-114.

[10] Chen, Yunnung, et al. (2016). End-to-End Memory Networks with Knowledge Carryover for Multi-Turn Spoken Language Understanding. Conference of the international speech communication association. 3245-3249.

[11] Su, Shangyu, Peichieh Yuan, and Yunnung Chen. (2018). How time matters: learning time-decay attention for contextual spoken language understanding in dialogues. North American chapter of the association for computational linguistics. 2133-2142.

[12] Bai, He, et al. (2019). Memory Consolidation for Contextual Spoken Language Understanding with Dialogue Logistic Inference. Conference of the association for computational linguistics. 5448-5453.

[13] Sankar, Chinnadhurai, et al. (2019). Do Neural Dialog Systems Use the Conversation History Effectively? An Empirical Study. Conference of the association for computational linguistics. 32-37.

[14] Kingma, D. P., & Ba, J. (2015). Adam: A Method for Stochastic Optimization. International conference on learning representations.