Overview of the NLPCC 2018 Shared Task: Spoken Language Understanding in Task-Oriented Dialog Systems

Xuemin Zhao[1(✉)] and Yunbo Cao[2(✉)]

[1] Tencent Technology (Chengdu) Company Limited, Chengdu, China
xueminzhao@tencent.com
[2] Tencent Technology (Beijing) Company Limited, Beijing, China
yunbocao@tencent.com

Abstract. This paper presents the overview for the shared task at the 7th CCF Conference on Natural Language Processing & Chinese Computing (NLPCC 2018): Spoken Language Understanding (SLU) in Task-oriented Dialog Systems. SLU usually consists of two parts, namely intent identification and slot filling. The shared task made publicly available a Chinese dataset of over 5.8 K sessions, which is a sample of the real query log from a commercial task-oriented dialog system and includes 26 K utterances. The contexts within a session are taken into consideration when a query within the session was annotated. To help participating systems correct ASR errors of slot values, this task also provides a dictionary of values for each enumerable type of slot. 16 teams entered the task and submitted a total of 40 SLU results. In this paper, we will review the task, the corpus, and the evaluation results.

Keywords: SLU · Intent identification · Slot filling

1 Introduction

In task-oriented dialog systems, understanding of users' queries (expressed in natural language) is a process of parsing users' queries and converting them into some structure that machine can handle. The understanding usually consists of two parts, namely intent identification and slot filling. For example, given the utterance "给我来一首谭咏麟的朋友", the user's intent is to play a song, and "谭咏麟" fills one slots (singer) and "朋友" fills another (song).

Intents are global properties of utterances, which signify the goal of a user. Slots, on the other hand, are local properties in the sense that they span individual words rather than whole utterances. And the words that fill slots tend to be the only semantically loaded words in the utterance (i.e., the other words are function words). In the dialog systems, each type of intent corresponds to a particular service API, and the slots correspond to the parameters required by the API. SLU helps the dialog system to call the right back-end service using the right parameters to satisfy users' goals.

Traditionally, both of intent identification and slot filling are considered one utterance at a time by the SLU process, and the context information (including both the

M. Zhang et al. (Eds.): NLPCC 2018, LNAI 11109, pp. 468–478, 2018.
https://doi.org/10.1007/978-3-319-99501-4_46

preceding queries in the same session and the user's situation information) is ignored by SLU and then handled by the dialog manager. The high cost to construct and maintain corpus is the main reason why the context information is not used in the SLU process. Usually, each utterance occurs within the context of a larger discourse between a person and a dialog system. Table 1 shows some example sessions, where without the context information from previous intra-session utterances we can't correctly do intent identification and slot filling for the utterance "取消" (utterance u_2 in session s_1, utterance u_2 in session s_2, utterance u_2 in session s_3) and "蒙曼" (utterance u_3 in session s_4). As the SLU process occurs in the early stage of a dialog system, well utilizing the context information can help avoid cascaded errors throughout the rest of the system.

Table 1. Example sessions, including session id **SID** and utterance ids **UID** in each session. Each utterance has an associated intent, while the corresponding slots are shown within each utterance using XML style tags.

SID	UID	Intent	Utterance
s_1	u_1	music.play	来一首<singer>冷漠</singer>的歌
s_1	u_2	music.pause	取消
s_2	u_1	navigation. navigation	导航去<destination>锡山紫金城</destination>
s_2	u_2	navigation. cancel_navigation	取消
s_3	u_1	phone_call. make_a_phone_call	呼叫<phone_num>4000008</phone_ num>
s_3	u_2	phone_call.cancel	取消
s_4	u_1	Others	说话
s_4	u_2	phone_call. make_a_phone_call	打电话给
s_4	u_3	phone_call. make_a_phone_call	<contact_name>蒙曼</contact_name>
s_4	u_4	navigation. navigation	<destination>增城宾馆</destination>
s_4	u_5	music.play	放音乐

Numerous techniques for SLU have been proposed, including traditional machine learning methods and hand-crafted features [1, 2, 4], deep learning methods [3, 5, 6], incorporating context information [1, 3], jointly optimizing intent detection and slot filling [5]. Despite this progress, direct comparisons between methods have not been possible because different datasets and domains are used in past studies.

The NLPCC 2018 Shared Task 4 (Spoken Language Understanding in Task-oriented Dialog Systems) provides a common testbed and evaluation suite for the SLU process. The shared task made publicly available a corpus of over 5.8 K sessions including 26 K utterances, which is a sample of the real query log from a commercial task-oriented dialog system. 16 teams entered the task, submitting a total of 40 SLU results.

This paper is organized as follows. First, Sect. 2 provides an overview of the task, the data and the evaluation metrics, all of which will remain publicly available to the community (NLPCC Shared Task 4, 2018). Then, Sect. 3 summarizes the results of the task. Finally, Sect. 4 briefly concludes.

2 Task Overview

2.1 Problem Statement

Spoken language understanding (SLU) comprises two tasks, intent identification and slot filling. That is, given the current query along with the previous queries in the same session, an SLU system predicts the intent of the current query and also all the slots associated with the predicted intent.

Included with the data is an ontology, which gives details of all the intents and the corresponding slots. To simplify the task, the dictionaries (e.g., singer, song, etc.) are

Table 2. Ontology and requirement in the task.

Intent	Slot	Provide-Slot-Dictionary	Do-Error-Correction
music.play	Song	YES	YES
	Singer	YES	YES
	Theme	YES	YES
	Style	YES	YES
	Age	YES	YES
	Toplist	YES	YES
	Emotion	YES	YES
	Language	YES	YES
	Instrument	YES	YES
	Scene	YES	YES
music.pause	–	–	–
music.prev	–	–	–
music.next	–	–	–
navigation. navigation	Destination	NO	NO
	custom_destination	YES	YES
	Origin	NO	NO
navigation.open	–	–	–
navigation. start_navigation	–	–	–
navigation. cancel_navigation	–	–	–
phone_call. make_a_phone_call	phone_num	NO	NO
	contact_name	NO	NO
phone_call.cancel	–	–	–
Others	–	–	–

provided for the slots with enumerable values while the slots with the non-enumerable values (e.g., phone_num, destination, contact_name, etc.) should be handled by rules or machine learning models. The textual strings, fed into a dialog system as input utterances, are mostly the transcripts translated from spoken language by ASR (Automatic Speech Recognition) and thus subject to recognition errors. If the enumerable slot values contain ASR errors, the SLU system should do slot value correction against the provided slot dictionaries. The non-enumerable slots don't need to do this for simplification. Table 2 gives details on the ontology used in this task.

The task studies the problem of SLU as a corpus-based task - i.e., the SLU systems are trained and tested on a static corpus of dialogs. The task is to re-run the SLU process on these dialogs - i.e., to take as input the dialogs translated from spoken language by ASR, and to output the SLU results. This corpus-based design was chosen because it allows different SLU systems to be evaluated on the same data.

2.2 Data

The dataset adopted by this task is a sample of the real query log from a commercial task-oriented dialog system, which is an in-car voice interface product. The data is all in Chinese. The evaluation includes three domains, namely music, navigation and phone call. Within the dataset, an additional domain label 'OTHERS' is used to annotate the data not covered by the three domains (as shown in Table 2). To simplify the task, we keep only the intents and the slots of high-frequency while ignoring others although they appear in the original data.

The entire data can be seen as a stream of user queries ordered by time stamp. The stream is further split into a series of segments according to the gaps of time stamps between queries and each segment is denoted as a "session". The annotation was achieved by first running an existing SLU system over the transcriptions, and then crowdsourcing to check the labels. Finally, the authors re-checked the labels by hand. The contexts within a session are taken into consideration when a query within the session was annotated. Table 1 gives some example sessions with annotations.

The entire dataset was randomly split into training and test dataset with a ratio of 4:1 at the session dimension. The statistics of the datasets are shown in Table 3. To help participating systems correct ASR errors, this task also provides a dictionary of values for each enumerable type of slot. Note that dictionaries are pruned such that they include all the values occurring in the dataset, but do not necessarily include all the values in real world. The statistics of the dictionaries are show in Table 4.

2.3 Evaluation

Depending on whether or not external resources can be used, the task can be divided into two types:

- Close evaluation – use only the training dataset provided by the task for model training and tuning, and output the results (in the evaluation stage) based only on the provided test set, not on any other dataset or resources.

- Open evaluation – can use any datasets and resources (in addition to the provided training dataset) for model training and tuning; and output the results (in the evaluation stage) based only on the provided test set, not no any other dataset or resources.

Table 3. The statistics of the datasets, where "# of" stands for "number of".

Item		Train dataset	Test dataset
# of sessions		4,705	1,177
# of utterances		21,352	5,350
Average session length		4.54	4.55
Average utterance length		5.93	6.08
# of error slot values		306	83
Intent	music.play	6,425	1,631
	music.pause	300	73
	music.prev	5	4
	music.next	132	34
	navigation.navigation	3,961	1,038
	navigation.open	245	55
	navigation.start_navigation	33	4
	navigation.cancel_navigation	835	207
	phone_call.make_a_phone_cal	2,796	670
	phone_call.cancel	22	18
	Others	6,598	1,616

Table 4. The statistics of the dictionaries.

Slot dictionary	Size
Song	6,870
Singer	2,667
Theme	140
Style	102
Age	139
Toplist	69
Emotion	135
Language	41
Instrument	30
Scene	145
custom_destination	3

Besides, we divided the task into another two sub-tasks: intent identification, and intent identification plus slot filling. In addition to the close and open evaluation, we got the following four sub-tasks:

- Sub-task 1: Intent Identification – Close;
- Sub-task 2: Intent Identification – Open;
- Sub-task 3: Intent Identification and Slot Filling – Close;
- Sub-task 4: Intent Identification and Slot Filling – Open.

However, it's very hard to do a close evaluation as the participating systems may use different Chinese word segmentor, word embedding, Name Entity Recognizer and dictionary resources. After the discussion with the participating teams, finally only Sub-task 2 and Sub-task 4 were retained in the final report, and Sub-task 1 and Sub-task 3 not.

For Sub-task 2, in order to balance the importance of each intent, we use $F1_{macro}$ of all the intents (not including the intent OTHERS) as the evaluation metric, calculated as the following equations,

$$P_{macro} = \frac{1}{N}\sum_{i=1}^{N} \frac{\#\text{ of queries correctly predicted as intent } c_i}{\#\text{ of queries predicted as intent } c_i},$$

$$R_{macro} = \frac{1}{N}\sum_{i=1}^{N} \frac{\#\text{ of queries correctly predicted as intent } c_i}{\#\text{ of queries labelled as intent } c_i},$$

$$F1_{macro} = \frac{2}{1/P_{macro} + 1/R_{macro}}.$$

For Sub-task 4, the evaluation metric is as given by the following equation,

$$P = \frac{\#\text{ of queries correctly parsed}}{\#\text{ of queries}},$$

where "# of queries" is the number of queries in the test set (including the queries with intent annotated as 'OTHERS'). "# of queries correctly parsed" denotes the number of queries for which the predicted intent and the predicted slot values (including the corrected values if correction is needed) are both exactly same as the annotations.

3 Results and Discussion

Altogether 16 teams participated in both of sub-tasks. Each team could submit a maximum of 3 results for each sub-task (Sub-task 2 and Sub-task 4), and both sub-tasks had 40 submitted entries in total. Table 5 gives the results on the metrics for each sub-task entry. As can be seen, the best result of Sub-task 2 is achieved by **AlphaGOU.entry3**, $F1_{macro} = 0.96157$; and the best result of Sub-task 4 is also achieved by **AlphaGOU.entry3**, P = 94.916% .

Table 5. Results of the evaluation.

Team ID	Sub-task 2			Sub-task 4	
	Entry	$F1_{micro}$	$F1_{macro}$	Entry	P
AlphaGOU	1	0.97090	0.96039	1	94.486%
	2	0.97234	0.96109	2	94.785%
	3	0.97365	**0.96157**	3	**94.916%**
CVTE_SLU	1	0.93390	0.92951	1	87.383%
	2	0.93454	0.93163	2	87.533%
	3	0.92675	0.91964	3	86.318%
DeepIntell	1	0.91659	0.60858	1	84.804%
	2	0.91942	0.67917	2	83.607%
	–	–	–	3	83.907%
DLUFL_SLU	1	0.93881	0.91612	1	88.112%
	2	0.94039	0.89501	2	88.710%
	3	0.93863	0.88936	3	88.243%
FAQRobot-wds	1	0.90584	0.86891	1	83.084%
	2	0.92594	0.91236	2	82.075%
	3	0.91481	0.88031	3	83.364%
HappyRogue	1	0.92785	0.76696	1	87.570%
	2	0.94211	0.89966	2	89.869%
	3	0.94105	0.89249	3	89.794%
HCCL	1	0.94873	0.92637	1	89.121%
	2	0.93211	0.91558	2	87.458%
	3	0.93339	0.91148	3	90.729%
ISCLAB	1	0.94474	0.85473	1	90.710%
laiye_rocket	1	0.93212	0.90285	1	79.813%
Learner	1	0.94886	0.91546	1	90.841%
	2	0.95197	0.93271	2	90.804%
	3	0.95223	0.94193	3	90.523%
orion_nlp	1	0.93065	0.88945	1	84.636%
	2	0.93038	0.90068	2	84.336%
	3	0.93035	0.88690	–	–
rax	1	0.93811	0.90409	1	88.168%
	2	0.93619	0.86398	2	87.028%
	3	0.93091	0.81014	3	86.430%
scau_SLU	1	0.94913	0.92989	1	78.374%
	2	0.94881	0.92962	2	78.486%
	3	0.94906	0.92972	3	79.720%
SLU-encoder	1	0.91981	0.87178	1	84.467%
	2	0.91978	0.87167	2	84.766%
	3	0.92023	0.87177	3	84.822%
SMIPG	1	0.91650	0.85222	1	82.972%

(*continued*)

Table 5. (*continued*)

Team ID	Sub-task 2			Sub-task 4	
	Entry	$F1_{micro}$	$F1_{macro}$	Entry	P
	2	0.91616	0.84256	2	82.916%
Team_4	1	0.85826	0.68953	1	74.785%

Table 5 also lists the metrics $F1_{micro}$ and $F1_{macro}$ for Sub-task 2. We could see that the metric $F1_{macro}$ is always less than the metric $F1_{micro}$ for all the entries. **CVTE_SLU.entry2** gets the least gap between $F1_{micro}$ and $F1_{macro}$, which is 0.00291. **DeepIntell.entry1** gets the greatest gap, which is 0.30801. The F1 metrics of all the intents for the two entries are shown in Fig. 1. In our released dataset, the example size of different intents is very different, and the maximum size is 895 times of the minimum. Because macro-averaging weights the metric toward the smaller classes, Teams should optimize the model performance for smaller classes (e.g. intents music.prev, navigation.start_navigation, and phone_call.cancel in Sub-task 2).

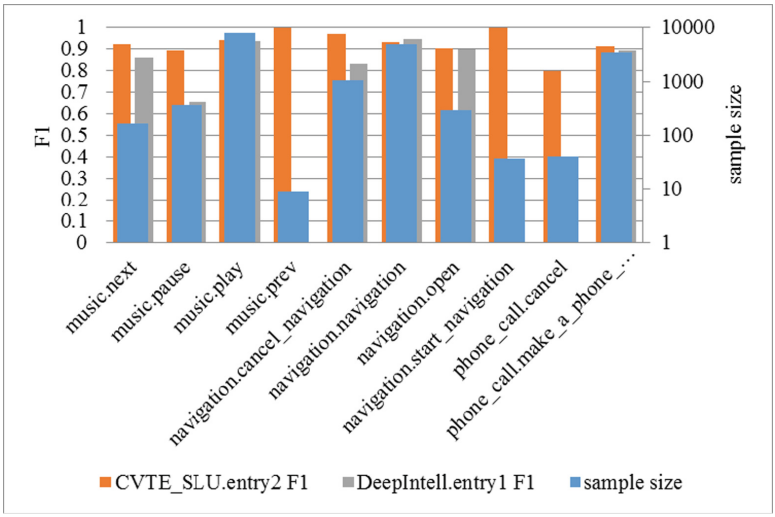

Fig. 1. Intent identification results of Sub-task 2 from **CVTE_SLU.entry2** and **DeepIntell. entry1**, and the sample size (including both of train and test datasets) for each intent. The F1 metrics for intents music.prev, navigation.start_navigation, and phone_call.cancel are all 0, and the sample size for these intents is 9, 37, and 40, respectively.

Figure 2 shows the results on slot filling (not combining the step of intent identification) of Sub-task 4 from **AlphaGOU.entry3**, which achieved the 1st place of Sub-task 4. Only the slots, whose sample size is larger than 100, are shown. One reason for the high performance of 'singer' and 'song' slots is that we released the slot dictionaries including all the values occurring in the dataset. The rich training data and

obvious features is the main reason for the high performance of the destination slot. The main reason for the relative low performance of the contact_name slot is that, firstly we didn't release the users' contact name lists because of the privacy protection, secondly the ASR performance of contact names is very poor.

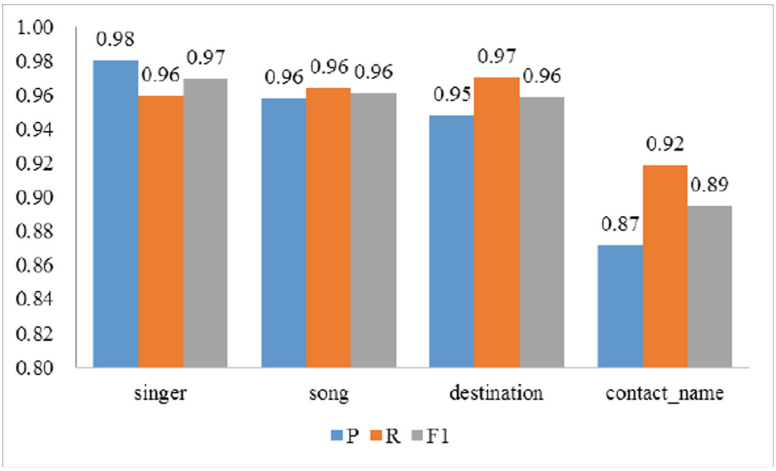

Fig. 2. Slot filling results (not combining the step of intent identification) of Sub-task 4 from **AlphaGOU.entry3**, where P stands for Precision, R for Recall, and F1 $= \frac{2}{1/P+1/R}$. The results are computed from the utterances whose intent identification is correct.

Figure 3 shows the results on slot value correction (not combining the steps of intent identification and slot filling) of Sub-task 4. We can see a big difference for the performance. The top right 3 points are given by the 3 entries of Team 1, who has achieved a precision of around 0.75 and a recall of around 0.76. 18 points lie in the bottom left corner (0, 0), which means that 18 entries from 8 teams didn't correct slot value errors.

3.1 Some Representative Systems

In this section, some representative systems will be briefly introduced. While most of the systems use the neural networks, the 1st places of the two sub-tasks are achieved by the **AlphaGOU** system using the traditional techniques.

AlphaGOU system is a hybrid of context-independent model and context-dependent rules; the former is a pipelined framework which includes slot boundary detection, slot type classification, slot correction and intent classifier. Although all the used techniques are very traditional, the system achieved promising results.

Learner system uses a hierarchical LSTM based model. The dialog history is memorized by a turn-level LSTM, which is used to assist the intent identification and slot filling.

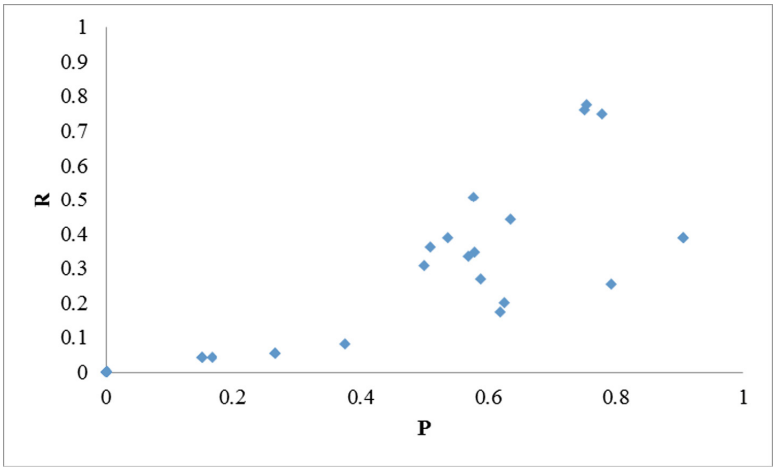

Fig. 3. Slot value correction results (not combining the steps of intent identification and slot filling) of Sub-task 4, where one point represents one entry result. P stands for Precision, and R stands for Recall.

ISCLAB system proposes a neural framework, named SI-LSTM model, which combines intent identification and slot filling together, and the slot information is used for determining the intent while the intent type is used to rectify the slot filling deviation.

4 Conclusion

In this paper, we present the overview of the NLPCC 2018 Shared Task: Spoken Language Understanding in Task-oriented Dialog Systems. The dataset adopted by this task is a sample of the real query log from a commercial task-oriented dialog system, which is an in-car voice interface product. The data is all in Chinese. The contexts within a session are taken into consideration when a query within the session was annotated. The entire dataset was randomly split into train and test dataset with a ratio of 4:1 at the session dimension. In the evaluation, two sub-tasks are designed. Sub-task 2 is intent identification, and Sub-task 4 is intent identification and slot filling. Both sub-tasks had 40 submitted entries in total. The best result of Sub-task 2 is achieved by **AlphaGOU.entry3**, $F1_{macro} = 0.96157$, and the best result of Sub-task 4 is also achieved by **AlphaGOU.entry3**, $\mathrm{P} = 94.916\%$.

Acknowledgement. We are very grateful to the colleagues from our company for their efforts to annotate the data. And we also would like to thank the participants for their valuable feedback.

References

1. Bhargava, A., Celikyilmaz, A., Hakkani-Tür, D., Sarikaya, R.: Easy contextual intent prediction and slot detection. In: ICASSP 2013 (2013)
2. Gong, N., Shen, T., Wang, T., Qi, D., Li, C.H.: The Sogou spoken language understanding system for the NLPCC 2018 evaluation. In: NLPCC 2018 (2018)
3. Xu, P., Sarikaya, R.: Contextual domain classification in spoken language understanding systems using recurrent neural network. In: ICASSP 2014 (2014)
4. Mairesse, F., et al.: Spoken language understanding from unaligned data using discriminative classification models. In: ICASSP 2009 (2009)
5. Xu, P., Sarikaya, R.: Convolutional neural network based triangular CRF for joint intent detection and slot filling. In: ASRU 2013 (2013)
6. Luo, B., Feng, Y., Wang, Z., Huang, S., Yan, R., Zhao, D.: Marrying up regular expressions with neural networks: a case study for spoken language understanding. In: ACL 2018 (2018)