The 1st Chinese Audio-Textual Spoken Language Understanding Challenge (CATSLU)

https://sites.google.com/view/CATSLU

Kai Yu, Tiejun Zhao, Chengqing Zong

April 8, 2019

Contents

1 Motivation

The spoken language understanding (SLU) is a key component of spoken dialogue system (SDS), parsing user's utterances into corresponding semantic concepts.

For example, the utterance *"Show me flights from Boston to New York"* can be parsed into *(fromloc.city_ name=Boston, toloc.city_ name=New York)*. Building a robust semantic parser system of multi-turn task-oriented spoken dialogue system is challenging, as it faces three main problems: variety of spoken language expression, uncertainty of automatic speech recognition (ASR) and adaption of dialogue domain.

Firstly, compared to written language, spoken language is much harder to be handled by a language understanding system. Since spoken language contains more complex linguistic phenomenons: the unnecessary repetitions, false starts, repairs and other disfluent situations. These phenomenons make it difficult to build a semantic parser system. Further more, spoken language processing always need automatic speech recognition (ASR) which converts speech to text. ASR errors make SLU become a challenging task. To improve robustness to ASR errors, audio information will be essential. A dialogue example is shown in Table 1.

Turn	Speaker	ASR 1-best	Transcription	Semantic annotation

1	user	给我点一首滚滚红尘
		Show me a song named Red Bust
			给我点一首滚滚红尘
			Show me a song named Red Bust
				inform(歌曲名=滚滚红尘)
				inform(musicname=Red Bust)

	Systerm	找到10首滚滚红尘，请问您要听哪个歌手的? There are ten musics named Red Bust, which edit do you like?		

2	user	第七首
		Seventh one, please.
			第七首
			Seventh one, please.
				inform(序列号=第七首)
				inform(serialnumber=seventh)

	Systerm	准备播放萨顶顶、常石磊的滚滚红尘。 Ready to play Sa and Chang's Red Bust.		

3	user	换一个歌曲
		Play another one, please.
			换一个歌曲
			Play another one, please.
				inform(操作=换一个)
				inform(operate=playanother)
				inform(对象=歌曲)
				inform(object=music)

	Systerm	请说歌手名字。 please inform a singer's name.		

4	user	比昂纳
		Bionna
			Beyond的
			Beyond
				inform(歌手名=Beyond)
				inform(singer=Beyond)

	Systerm	请问您要听谁的歌? Who's song do you want to listen?		

5	user	黄阳家具
		Huang Yang furniture
			黄家驹
			Wong Ka Kui
				inform(歌手名=黄家驹)
				inform(singer=Wong Ka Kui)

	Systerm	再见! Bye!		

Table 1: Example of an ASR 1-best (top hypothesis), transcription and semantic annotation for a dialogue in the "music searching" domain. It shows how speech recognition errors destroy spoken language understanding.

Secondly, it is also hard to obtain enough labelled data of new dialogue domain, since data collection and annotation in the flow of dialogue are very expensive and time-consuming. Therefore, domain adaptation of SLU became important, which tries to train a semantic parser on some source domains and then adapt it to the target domain. Different from the music domain in Table 1, a dialogue example of video domain is shown in Table 2.

To fully investigate these problems and promote application of spoken dialogue system, we will release a multi-turn task-oriented Chinese spoken dialogue dataset (as shown in Table 3) and organize the first Chinese Audio-Textual Spoken Language Understanding Challenge (CATSLU). This challenge consists of two sub-challenges:

Turn	Speaker	ASR 1-best	Transcription	Semantic annotation

1	user	放个动画片
		Play a cartoon, please.
			放个动画片
			Play a cartoon, please.
				inform(对象=动画片)
				inform(object=cartoon)

	Systerm	请问您要看什么动画片？ Which cartoon do you like?		

2	user	小猪佩奇
		Peppa Pig
			小猪佩奇
			Peppa Pig
				inform(片名=小猪佩奇)
				inform(name=Peppa Pig)

	Systerm	准备动画片小猪佩奇。 Ready to play Peppa Pig.		

Table 2: Example of an ASR 1-best (top hypothesis), transcription and semantic annotation for a dialogue in the "video searching" domain.

1. SLU in single domain: Perform a slot filling system in a single domain. A large number of training dialogues related to **music** search and **map** navigation will be released (20% utterances will be randomly selected as test data). The data was collected from dialogues between users and a manageable spoken dialogue system (human-computer interaction), which happened in the real world. Both audio and text information are very important for understanding users. Therefore, audio features will also be provided as well as text features.

2. Domain adaptation of SLU: Adapt the SLU model of source domain to target domain. We will set **music** and **map** domain as source domains, while leave **video** and **weather** as target domains (20% utterances will be randomly selected as seed data and the rest is used for evaluation). Participants can use the seed data plus the **music** and **map** data from the first sub-challenge for adaptive training.

Domain		#Users	#Utterances	#Slot
source	Map	1788	7592	24
	Music	268	3246	20
target	Weather	276	3379	22
	Video (film and TV)	227	2041	28

Table 3: Statistics of CATSLU dataset.

There are some previous research challenges (like DSTC 2&3 [1, 2] [1]) for spoken language understanding tasks, which are similar to this challenge. But our dataset is more realistic with more domains and a large number of slot-values. It is collected from the real-world SDS application supported by AISPEECH[2] devoted in research and application of human-computer dialogue technology.

2 Participation

In this challenge, participants will be provided with labelled human-computer dialogues to develop spoken language understanding systems. Systems will then be evaluated on a common set of held-out dialogues by the organizers to enable

[1] http://camdial.org/~mh521/dstc/

[2] http://www.aispeech.com/index.php

fair comparisons. This challenge focuses on robust SLU based on both audio and textual inputs. It is encouraged to exploit audios (to build a speech recognizer or used as multimodal features) but not only text.

As well as a corpus of labelled dialogues, participants will be given code that implements the evaluation measurements and a baseline SLU parser.

For evaluations, participants will be given an unlabelled mini-test set to make sure their systems can be executed, offline, on test set. Participants will submit their SLU systems to the organizers, who will then perform the evaluation on the unlabelled test set in a week. After evaluation, the scoring results and submitted output of all participants will also be made public. If making your system's output public would prevent your team from participating, please contact the organizers.

Participants are provided with:

1. Training and development data: Annotated log files in a common JSON format, ontology file, lexicon files and the corresponding audio files.
2. Mini test set without labels.
3. The scoring tool that will be used to evaluate the SLU systems.
4. Two baseline SLU systems.

2.1 Rules

Participation is welcomed from any research team (academic, corporate, nonprofit, government). Members of the organizational committee and technical committee are permitted to participate. In general, the identity of participants will not be published or made public. In written results, teams will be identified as team1, team2, etc. There are 2 exceptions to this: (1) the organizers will verbally indicate the identities of all teams at the conference/workshop chosen for communicating results; and (2) participants may identify their own team label (e.g. team5), in publications or presentations, if they desire, but may not identify the identities of other teams.

On submission of systems, teams will be required to fill in a questionnaire which gives some broad details about the approach they adopted.

2.2 System Submission

For all participants, they can follow the instructions below to submit the systems:

- Participants can contribute in all sub-challenges or just one at a time;
- Participants can submit and update their systems until the final system submission deadline;
- Make sure your system can be executed offline in Linux.
- Each participant has up to five submission attempts per domain (*map*, *music*, *weather*, *video*);
- The best result out of the submissions will be used to determine the winner of each domain.

And the specification of result submission is as follows:

- Participants can email their systems in a zip file to 1248uu@sjtu.edu.cn, paul2204@sjtu.edu.cn and zhenchi713@sjtu.edu.cn.
- The subject of the email should be "Systems_TeamName". The zip file should include the name of the team, the focused domain (*map*, *music*, *weather*, *video*), and the number of this attempt (from 1 to 5), e.g. "Systems_SJTU_map_1.zip".

2.3 Paper Submission

A paper submission and at least one upload on the test set are mandatory for the participation in the Challenge. However, paper contributions within the scope are also welcome if the authors do not intend to participate in the Challenge itself. In any case, researchers should submit their papers until 5th July 2019 using the standard style info and respecting length limits.

The papers will undergo the normal review process. Papers should refer to the baseline paper for details about the dataset and baseline results. This makes for a more readable set of papers, compared to each challenge paper repeating the same information. Please cite the introductive paper which will be available soon.

All papers must be formatted according to ICMI proceedings style, and should be no more than 4 pages in the two-column ACM conference format (excluding references). Papers should be submitted through the ICMI challenge's easychair submission site (submission link will be made available soon). Reviewing will be double-blind, so submissions should be anonymous: do not include the authors' names, affiliations or any clearly identifiable information in the paper. Latex and word templates for this format can be downloaded from the main website.

2.4 Schedule

Our expected schedule for the challenge is shown in Table 4.

Date	Description
4th April	Release of labelled training, development sets and minimal test (unlabelled) samples.
10th April	Release of prelimary baseline paper.
20th June	Closing competition.
20th June	Final system submission.
21-27th June	Running submitted systems and evaluating results on the unlabelled test sets by the organizers.
28th June	Results are given back to the participants.
5th July	Paper submission deadline.
31st July	Paper Notification.
12th August	Camera ready.

Table 4: Expected schedule for the challenge.

3 Data

This section describes the data released for CATSLU challenge. Datasets consist of the user responses or utterances, which are collected from four dialog domains (*map*, *music*, *weather*, *video*). The transcription and semantics of the user utterance are annotated. For each domain, the labelled utterance objects used to train SLU models are distributed in a JSON files, named **train.json**. We also provide the development set **development.json** for validation and keep the test set **test.json** for evaluation. Each domain is described by an ontology object (also distributed in JSON file). The formats of these objects are fully specified in Appendix B.

The ontology of each domain contains all possible *acts*, *slots* and *values* which would exist in this domain. Specifically, some slots could be constrained by a value, but the others are not, as shown in Table 5. In this challenge, we assume all possible values of a slot are known. The number of possible values of a slot could be huge in the real world, e.g. the number of addresses must be millions. To facilitate the challenge, we limit the value set of each slot to a small scale.

Map		Music		Weather		Video	
Slot	Constraint	Slot	Constraint	Slot	Constraint	Slot	Constraint
poi名称	✓(4976)	专辑名	✓(97)	国家	✓(194)	国家地区	✓(165)
poi修饰	✓(4976)	歌曲名	✓(1196)	省份	✓(58)	演员	✓(124)
poi目标	✓(4976)	歌手名	✓(396)	城市	✓(810)	电影人	✓(327)
起点名称	✓(4976)	主题	✓(108)	区域	✓(5792)	片名	✓(558)
起点修饰	✓(4976)	乐器	✓(47)	地点	✓(504)	导演	✓(188)
起点目标	✓(4976)	语种	✓(56)	风景点	✓(501)	影视标签	✓(104)
终点名称	✓(4976)	适用人名	✓(449)	日期	✓(957)	影视类型	✓(85)
终点修饰	✓(4976)	适用人群	✓(72)	时间	✓(8678)	排行榜	✓(9)
终点目标	✓(4976)	适用年龄	✓(45)	气象	✓(57)	播放时间	✓(3)
途经点名称	✓(4976)	音乐场景	✓(50)	温度	✓(7)	清晰度	✓(19)
请求类型	✓(7)	音乐类型	✓(134)	湿度	✓(4)	电影奖	✓(118)
出行方式	✓(1)	音乐风格	✓(119)	空气质量	✓(13)	系列	✓(32)
路线偏好	✓(10)	主题曲类型	✓(3)	场景	✓(6)	视频源	✓(9)
对象	✓(12)	应用名	✓(16)	活动	✓(18)	角色名	✓(23)
操作	✓(46)	对象	✓(6)	装备	✓(11)	评分	✓(99)
序列号	✓(94)	操作	✓(22)	服装	✓(24)	评分来源	✓(8)
页码	✓(2)	序列号	✓(74)	PM2.5	×	语种	✓(56)
value	✓(1)	页码	✓(2)	具体时间	×	适用人名	✓(445)
位置	×	歌曲数量	✓(5)	天气	×	适用人群	✓(72)
前方路况	×	value	✓(1)	最低温度	×	集数	✓(41)
剩余时间	×	-	-	最高温度	×	上映时间	✓(5)
剩余距离	×	-	-	穿衣指数	×	应用名	✓(8)
剩余路程	×	-	-	-	-	对象	✓(22)
路况	×	-	-	-	-	操作	✓(22)
-	-	-	-	-	-	序列号	✓(54)
-	-	-	-	-	-	页码	✓(3)
-	-	-	-	-	-	影视数量	✓(6)
-	-	-	-	-	-	value	✓(1)

Table 5: Slots of each domain. When a slot can be constrained, there is a number of all the possible values.

4 Evaluation

A SLU system for one domain should generate the SLU output for every utterance in a given JSON log file described in Section 3. The audios, manual transcriptions, ASR 1-best hypothesis, and even lexicons can be used in the training stage, while manual transcriptions are forbidden in the evaluation. Moreover, any information from the future turns in a dialogue are not allowed to be considered to parsing the semantic items at a given turn.

4.1 SLU output

A SLU outputs for each utterance is a set of semantic items. The semantic item is a tuple which has two or three elements. The elements can be classified into *act*, *slot* or *value*.

- The semantic item is a tuple with three elements. It is tuple (*act*, *slot*, *value*) to provide information, e.g. *(inform, date, today)*, *(deny, song, my heart will go on)*, *(confirm, destination, Shanghai)* etc.
- The semantic item is a tuple with two elements. It is tuple (*act*, *slot*) to ask for values, e.g. *(request, date)*, *(request, destination)* etc.

4.2 Metrics

There are two SLU performance that are measured, using the following metrics:

- **Accuracy**: fraction of utterances in which the predicted semantic items are totally correct.
- Precision/Recall/F-score:
  - **Precision**: fraction of semantic items in the SLU outputs that are correctly predicted.
  - **Recall**: fraction of semantic items in the gold standard labels that are correctly predicted.
  - **F-score**: the harmonic mean of precision and recall.

5 Included Scripts and Tools

This challenge comes with a download which packages some useful scripts and tools for dealing with the data.

5.1 Important Scripts

- **scripts/score.py**: a scoring script to compute the metrics of SLU performance, which will be used in the evaluation stage.
- **scripts/ontology.py**: contains a function to read domain ontology and also the related lexicon files.

5.2 Ruler Baseline

This baseline system works in a simple string matching. As all the possible values of each slot are known in the ontology, so we apply a string matching between values and the text features of utterance to make a SLU prediction.

You can run the ruler baseline like so:

```
python3 ruler/lexicon_matching.py --domain map
    --dataset development --dataroot data/map
    --text_type asr_1best --output output.json
```

This will create a file `output.json` with a SLU output object. The structure and contents of the output are just similar to the input. The evaluation script, `scripts/score.py` can be run on the SLU output like so:

```
python3 scripts/score.py --prediction output.json
     --annotation data/map/development.json
```

This prints all the metrics described in Section 4.2.

```
+----------------+------------+
|    metrics     | scores (%) |
+----------------+------------+
|   Precision    |   26.67    |
|     Recall     |   87.18    |
|    F1-score    |   40.84    |
| Joint accuracy |   42.78    |
+----------------+------------+
```

Finally, we will test SLU performance on unlabelled test set like so:

```
python3 ruler/lexicon_matching.py --domain map --dataset
     test_unlabelled --dataroot data/map --text_type
     asr_1best --output output.json
```

Only minimal samples are provided to participants in `test_unlabelled.json` of each domain. In the evaluation stage, the organizers will fill the `test_unlabelled.json` of each domain with full test data.

We also prepare a demo script `demo_lexicon_matching.sh` for all domains.

5.3 Neural Network based Baseline

This baseline system adopt the newly proposed hierarchical decoding model [3]. The model dynamically parses act, slot and value in a structured way and employs pointer network to handle out-of-vocabulary (OOV) values. You can follow the instructions below to run this baseline and read the `neuralNet/README.md` for details.

First of all, we need to preprocess the data, for example, word segmentation and vocabulary construction. In this baseline, we just split the sentence into chars without special tokenizer. Participators can try complex tokenization methods with utilization of ontology.

```
mkdir processed_data
python3 neuralNet/preprocess.py -utt_mode asr_1best
     -train_file data/map/train.json
```

```
    -valid_file data/map/development.json
    -save_dir processed_data -min_freq 1
```

Next, we can train the model. We train up to 50 epochs (it takes about one minute per epochs on GTX 1080 ti) and save the model performing best in validation data.

```
mkdir exp_asr_1best
python3 neuralNet/train.py -data_root processed_data
    -experiment exp_asr_1best
```

Then, we can test the model as follows:

```
python3 neuralNet/test.py  -utt_mode asr_1best
    -test_file data/map/development.json
    -data_root processed_data
    -load_model exp_asr_1best/best.pt
    -save_file exp_asr_1best/output.json
```

Finally, we can evaluate the SLU output like so:

```
python3 scripts/score.py
    --prediction exp_asr_1best/output.json
    --annotation data/map/development.json
```

This prints all the metrics described in Section 4.2.

```
+-----------------+------------+
|     metrics     | scores (%) |
+-----------------+------------+
|    Precision    |   77.96    |
|      Recall     |   76.80    |
|     F1-score    |   77.37    |
|  Joint accuracy |   72.64    |
+-----------------+------------+
```

Evaluation on unlabelled test set is just like that on validation set. We also prepare a demo script `demo_NN_parser.sh` for all domains.

6 Committees

6.1 Organizing Committee

The organizers of the challenge are as follows:

- **Kai Yu** - kai.yu@sjtu.edu.cn, Chair of SpeechLab, Computer Science and Engineering Department, Shanghai Jiao Tong University, China.

- **Tiejun Zhao** - tjzhao@hit.edu.cn, Professor of Research Center of Language Technology, Deputy Director of MOE-MS Key Laboratory of NLP & Speech, School of Computer Science and Technology, Harbin Institute of Technology, China.

- **Chengqing Zong** - cqzong@nlpr.ia.ac.cn , Researcher, National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences, China.

6.2 Program Committee

The technical committee members of the challenge are as follows:

- **Masao Utiyama**, Research Manager of the National Institute of Information and Communications Technology, Japan.
- **Milica Gasic**, Group Lead at Saarland University, Germany.
- **Juanzi Li**, Principal of Knowledge Engineering Group, Tsinghua University, China.
- **Min Zhang**, Director of Institute of Human Language Technology, Soochow University, China.
- **Xuanjing Huang**, Professor of the School of Computer Science, Fudan University, China.
- **Kallirroi Georgila**, Research Assistant Professor of the Department of Computer Science, University of Southern California, USA.
- **Kees van Deemter**, Professor of the Department of Information and Computing Sciences, Utrecht University, Holland.

A File Directory Structure

In this section, the file directory structure of the CATSLU challenge will be introduced in detail. Actually, only a few of samples are provided in `test_unlabelled.json`. The rest of unlabelled test data is kept by the organizers.

```
catslu_traindev
├── data
│   ├── map
│   │   ├── audios
│   │   │   ├── 001bfd1d174c187e3e_59bf67fc332009c.wav
│   │   │   └── ...
│   │   ├── lexicon
│   │   │   ├── poi_name.txt
│   │   │   └── ...
│   │   ├── development.json
│   │   ├── train.json
│   │   ├── test_unlabelled.json
│   │   └── ontology.json
│   ├── music
│   │   ├── audios/
│   │   ├── lexicon/
│   │   ├── development.json
│   │   ├── train.json
│   │   ├── test_unlabelled.json
│   │   └── ontology.json
│   ├── video
│   │   ├── audios/
│   │   ├── lexicon/
│   │   ├── development.json
│   │   ├── train.json
│   │   ├── test_unlabelled.json
│   │   └── ontology.json
│   └── weather
│       ├── audios/
│       ├── lexicon/
│       ├── development.json
│       ├── train.json
│       ├── test_unlabelled.json
│       └── ontology.json
├── ruler/
├── neuralNet/
├── scripts
│   ├── ontology.py
│   └── score.py
├── demo_lexicon_matching.sh
├── demo_NN_parser.sh
└── README.md
```

B JSON Data Formats

The labelled dialogues in the development set and the training set are recorded in the JSON files, named **development.json** and **train.json** respectively. For each dialogue domain, there is an Ontology JSON object (recorded in **ontology.json**), which describes the corresponding ontology/domain of the dialogues. The below sections describe the structures of the labelled dialogue object and ontology object.

The JSON structures are specified by giving the field in a **bold type-face**. Fields which map to an object produce an indented list of the fields contained in that object. Fields mapping to lists of one type of object, will again have that object specified in an indented list surrounded by square brackets.

In order to make this clear, we present an example specification and object. The below is a specification to describe a music object:

- **title**: (string)
- **date**: (time)
- **authors**: [
  - **first name**: (string)
  - **last name**: (string)

  ]
- **nationality**: (string)

An example JSON book object conforming to this specification would then be:

```
{
  "title": "歌唱祖国(Song of Motherland)",
  "date": 9/12/1951,
  "authors": [
    {
      "first name": "莘(Xin)",
      "last name": "王(Wang)"
    }
  ],
  "nationality": "中国(China)"
}
```

B.1 Label Objects

Every dialogue has been annotated, saved in the files called **development.json** and **train.json**, following the below specification:

- **dlg_id**: a unique ID for this dialogue (string)
- **utterances**: a list of the labelled dialogue turns, which give the response captured from the user. [

  - **utt_id**: the turn number of the dialogue. (integer)
  - **wav_id**: a unique ID for this utterance audio, which is the corresponding filename in the file named **audios**. (string)
  - **manual_transcript**: annotated by human according to the audio, which wouldn't be provided in test set. (string)
  - **asr_1best**: the best result by the ASR model. (string)
  - **semantic**: the semantic label by experts. (list) [[
    - **action**: dialogue action. (string),
    - **slot**: the slot mentioned in this dialogue action. (string),
    - **value**: the specific value of the slot. (string)

    ]]

]

B.2 Ontology Objects

An Ontology object describes the ontology (or domain) of a set of dialogues, in the following format:

- **acts**: the list of allowed possible values for the dialogue action. There isn't a value, when the action is labelled by 0. (object)
- **informable**: a mapping from a slot to a set of all possible values when the slot can be informed by the user (i.e. give as a constraint). If the count of the value set is large, all the slot values are distributed in a text file. In this situation, instead of directly presenting all the values of the slot, a slot-value text file path is given. The text file is a lexicon file in which each line is a possible value. (object)
- **requestable**: a list of slots which may be requested by the user. (list)

References

[1] Matthew Henderson, Blaise Thomson, and Jason D Williams, "The second dialog state tracking challenge," in *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, 2014, pp. 263–272.

[2] Matthew Henderson, Blaise Thomson, and Jason D Williams, "The third dialog state tracking challenge," in *2014 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2014, pp. 324–329.

[3] Zijian Zhao, Su Zhu, and Kai Yu, "A hierarchical decoding model for spoken language understanding from unaligned data," *arXiv preprint arXiv:1904.04498*, 2019.