ConvLab-2: An Open-Source Toolkit for Building, Evaluating, and Diagnosing Dialogue Systems

Qi Zhu[†] **Zheng Zhang**[†] **Yan Fang**[†] **Xiang Li**[†] **Ryuichi Takanobu**[†]
Jinchao Li[‡] **Baolin Peng**[‡] **Jianfeng Gao**[‡] **Xiaoyan Zhu**[†] **Minlie Huang**[†]
[†]Tsinghua University, China [‡]Microsoft Research, USA
[†]{zhu-q18,z-zhang15,fangy17,gxly19}@mails.tsinghua.edu.cn
[‡]{jincli,bapeng,jfgao}@microsoft.com [†]{zxy-dcs,aihuang}@tsinghua.edu.cn

Abstract

We present *ConvLab-2*, an open-source toolkit that enables researchers to build task-oriented dialogue systems with state-of-the-art models, perform an end-to-end evaluation, and diagnose the weakness of systems. As the successor of ConvLab (Lee et al., 2019b), ConvLab-2 inherits ConvLab's framework but integrates more powerful dialogue models and supports more datasets. Besides, we have developed an analysis tool and an interactive tool to assist researchers in diagnosing dialogue systems. The analysis tool presents rich statistics and summarizes common mistakes from simulated dialogues, which facilitates error analysis and system improvement. The interactive tool provides a user interface that allows developers to diagnose an assembled dialogue system by interacting with the system and modifying the output of each system component.

1 Introduction

Task-oriented dialogue systems are gaining increasing attention in recent years, resulting in a number of datasets (Henderson et al., 2014; Wen et al., 2017; Budzianowski et al., 2018b; Rastogi et al., 2019) and a wide variety of models (Wen et al., 2015; Peng et al., 2017; Lei et al., 2018; Wu et al., 2019; Gao et al., 2019). However, very few open-source toolkits provide full support to assembling an end-to-end dialogue system with state-of-the-art models, evaluating the performance in an end-to-end fashion, and analyzing the bottleneck both qualitatively and quantitatively. To fill the gap, we have developed **ConvLab-2** based on our previous dialogue system platform ConvLab (Lee et al., 2019b). ConvLab-2 inherits its predecessor's framework and extend it by integrating many recently proposed state-of-the-art dialogue models. In addition, two powerful tools, namely the analysis tool and the interactive tool, are provided for in-depth error analysis.

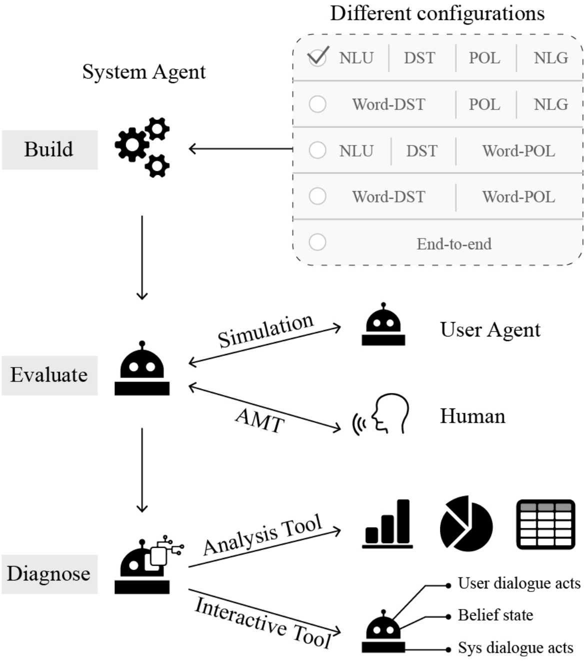


Figure 1: Framework of ConvLab-2. The top block shows different approaches to build a dialogue system.

As shown in Figure 1, there are many approaches to building a task-oriented dialogue system, ranging from pipeline methods with multiple components to fully end-to-end models. Previous toolkits focus on either end-to-end models (Miller et al., 2017) or one specific component such as dialogue policy (POL) (Ultes et al., 2017), while the others toolkits that are designed for developers (Bocklisch et al., 2017; Papangelis et al., 2020) do not have state-of-the-art models integrated. ConvLab (Lee et al., 2019b) is the first toolkit that provides various powerful models for all dialogue components and allows researchers to quickly assemble a complete dialogue system (using a set of recipes). ConvLab-2 inherits the flexible framework of ConvLab and imports recently proposed models that achieve state-of-the-art performance. In addition, ConvLab-2 supports several large-scale dialogue datasets including CamRest676 (Wen et al., 2017), MultiWOZ (Budzianowski et al., 2018b), DealOrNoDeal (Lewis et al., 2017), and CrossWOZ (Zhu et al., 2020).

To support an end-to-end evaluation, ConvLab-2 provides user simulators for automatic evaluation and integrates Amazon Mechanical Turk for human evaluation, similar to ConvLab. Moreover, it provides an analysis tool and an human-machine interactive tool for diagnosing a dialogue system. Researchers can perform quantitative analysis using the analysis tool. It presents useful statistics extracted from the conversations between the user simulator and the dialogue system. This information helps reveal the weakness of the system and signifies the direction for further improvement. With the interactive tool, researchers can perform qualitative analysis by deploying their dialogue systems and conversing with the systems via the webpage. During the conversation, the intermediate output of each component in a pipeline system, such as the user dialogue acts and belief state, are presented on the webpage. In this way, the performance of the system can be examined, and the prediction errors of those components can be corrected manually, which helps the developers identify the bottleneck component. The interactive tool can also be used to collect real-time human-machine dialogues and user feedback for further system improvement.

2 ConvLab-2

2.1 Dialogue Agent

Each speaker in a conversation is regarded as an agent. ConvLab-2 inherits and simplifies ConvLab's framework to accommodate more complicated dialogue agents (e.g., using multiple models for one component) and more general scenarios (e.g., multi-party conversations). Thanks to the flexibility of the agent definition, researchers can build an agent with different types of configurations, such as a traditional pipeline method (as shown in the first layer of the top block in Figure 1), a fully end-to-end method (the last layer), and between (other layers) once instantiating corresponding models. Researchers can also freely customize an agent, such as incorporating two dialogue systems into one agent to cope with multiple tasks. Based on the unified agent definition that both dialogue systems and user simulators are treated as agents, ConvLab-2 supports conversation between two agents and can be extended to more general scenarios involving three or more parties.

2.2 Models

ConvLab-2 provides the following models[1] for every possible component in a dialogue agent. Note that compared to ConvLab, newly integrated models in ConvLab-2 are marked in bold. Researchers can easily add their models by implementing the interface of the corresponding component. We will keep adding state-of-the-art models to reflect the latest progress in task-oriented dialogue.

2.2.1 Natural Language Understanding

The natural language understanding (NLU) component, which is used to parse the other agent's intent, takes an utterance as input and outputs the corresponding dialogue acts. ConvLab-2 provides three models: Semantic Tuple Classifier (STC) (Mairesse et al., 2009), MILU (Lee et al., 2019b), and **BERTNLU**. BERT (Devlin et al., 2019) has shown strong performance in many NLP tasks. Thus, ConvLab-2 proposes a new BERTNLU model. BERTNLU adds two MLPs on top of BERT for intent classification and slot tagging, respectively, and fine-tunes all parameters on the specified tasks. BERTNLU achieves the best performance on MultiWOZ in comparison with other models.

2.2.2 Dialogue State Tracking

The dialogue state tracking (DST) component updates the belief state, which contains the constraints and requirements of the other agent (such as a user). ConvLab-2 provides a rule-based tracker that takes dialogue acts parsed by the NLU as input.

2.2.3 Word-level Dialogue State Tracking

Word-level DST obtains the belief state directly from the dialogue history. ConvLab-2 integrates three models: MDBT (Ramadan et al., 2018), **SUMBT** (Lee et al., 2019a), and **TRADE** (Wu et al., 2019). TRADE generates the belief state from utterances using a copy mechanism and achieves state-of-the-art performance on MultiWOZ.

2.2.4 Dialogue Policy

Dialogue policy receives the belief state and outputs system dialogue acts. ConvLab-2 provides a rule-based policy, a model-based policy where the model can be learned via supervised learning,

[1] Please refer to the ConvLab-2 site for more details.

or reinforcement learning methods such as REINFORCE (Williams, 1992), PPO (Schulman et al., 2017), and **GDPL** (Takanobu et al., 2019). GDPL applies Adversarial Inverse Reinforcement Learning for joint reward estimation and policy optimization and achieves state-of-the-art performance on MultiWOZ.

2.2.5 Natural Language Generation

The natural language generation (NLG) component transforms dialogue acts into a natural language sentence. ConvLab-2 provides a template-based method and SC-LSTM (Wen et al., 2015).

2.2.6 Word-level Policy

Word-level policy directly generates a natural language response (rather than dialogue acts) according to the dialogue history and the belief state. ConvLab-2 integrates three models: MDRG (Budzianowski et al., 2018a), **HDSA** (Chen et al., 2019), and **LaRL** (Zhao et al., 2019). MDRG is the baseline model proposed by Budzianowski et al. (2018b) on MultiWOZ, while HDSA and LaRL achieve much stronger performance on this dataset.

2.2.7 User Policy

User policy is the core of a user simulator. It takes a pre-set user goal and system dialogue acts as input and outputs user dialogue acts. ConvLab-2 provides an agenda-based (Schatzmann et al., 2007) model and several neural network-based models including HUS and its variational variants (Gür et al., 2018). To perform end-to-end simulation, researchers can equip the user policy with NLU and NLG components to assemble a complete user simulator.

2.2.8 End-to-end Model

A fully end-to-end dialogue model receives the dialogue history and generates a response in natural language directly. ConvLab-2 extends Sequicity (Lei et al., 2018) to multi-domain scenarios: when the model senses that the current domain has switched, it resets the belief span, which records information of the current domain. As for the DealOrNoDeal dataset, we have also implemented the **ROLLOUTS RL** policy, which is based on the REINFORCE algorithm (Williams, 1992) and uses Monte Carlo tree search to choose the best action to respond that maximizes the reward.

2.3 Datasets

Compared with ConvLab, ConvLab-2 can integrate a new dataset more conveniently. For each dataset, ConvLab-2 provides a unified data loader that can be used by all the models, thus separating data processing from the model definition. Currently, ConvLab-2 supports four task-oriented dialogue datasets, including CamRest676 (Wen et al., 2017), MultiWOZ (Eric et al., 2019), DealOrNoDeal (Lewis et al., 2017), and CrossWOZ (Zhu et al., 2020).

2.3.1 CamRest676

CamRest676 (Wen et al., 2017) is a Wizard-of-Oz dataset, consisting of 676 dialogues in a restaurant domain. ConvLab-2 offers an agenda-based user simulator and a complete set of models for building a traditional pipeline dialogue system on the CamRest676 dataset.

2.3.2 MultiWOZ

MultiWOZ (Budzianowski et al., 2018b) is a large-scale multi-domain Wizard-of-Oz dataset. It consists of 10,438 dialogues with system dialogue acts and belief states. However, user dialogue acts are missing, and belief state annotations and dialogue utterances are noisy. To address these issues, Convlab (Lee et al., 2019b) annotated user dialogue acts automatically using heuristics. Eric et al. (2019) further re-annotated the belief states and utterances, resulting in the MultiWOZ 2.1 dataset.

2.3.3 DealOrNoDeal

DealOrNoDeal (Lewis et al., 2017) is a dataset of human-human negotiations on a multi-issue bargaining task. It contains 5,805 dialogues based on 2,236 unique scenarios. On this dataset, ConvLab-2 implements ROLLOUTS RL (Lewis et al., 2017) and LaRL (Zhao et al., 2019) models.

2.3.4 CrossWOZ

CrossWOZ (Zhu et al., 2020) is the first large-scale Chinese multi-domain Wizard-of-Oz dataset proposed recently. It contains 6,012 dialogues spanning over five domains. About 60% of the user goals have inter-domain dependencies that imply transitions across domains. Besides dialogue acts and belief states, the annotations of user states, which indicate the completion of a user goal, are also provided. ConvLab-2 offers a rule-based user simulator and a complete set of models for building a pipeline system on the CrossWOZ dataset.

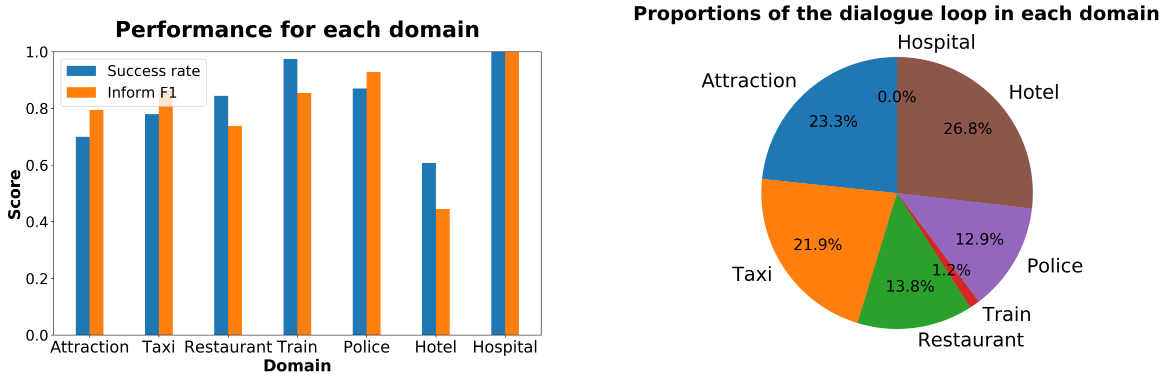


Figure 2: Performance of the demo system in Section 3. **Left:** Success rate and inform F1 for each domain. **Right:** Proportions of the dialogue loop in each domain.

Overall results:
Success Rate: 60.8%; inform F1: 44.5%

Most confusing user dialogue acts:
Request-Hotel-Post-?
- 34%: Request-Hospital-Post-?
- 32%: Request-Attraction-Post-?

Request-Hotel-Addr-?
- 29%: Request-Attraction-Addr-?
- 28%: Request-Restaurant-Addr-?

Request-Hotel-Phone-?
- 26%: Request-Restaurant-Phone-?
- 26%: Request-Attraction-Phone-?

Invalid system dialogue acts:
- 31%: Inform-Hotel-Parking
- 28%: Inform-Hotel-Internet

Redundant system dialogue acts:
- 34%: Inform-Hotel-Stars

Missing system dialogue acts:
- 25%: Inform-Hotel-Phone

Most confusing system dialogue acts:
Recommend-Hotel-Parking-yes
- 21%: Recommend-Hotel-Parking-none
- 18%: Inform-Hotel-Parking-none

Inform-Hotel-Parking-yes
- 17%: Inform-Hotel-Parking-none

Inform-Hotel-Stars-4
- 16%: Inform-Hotel-Internet-none

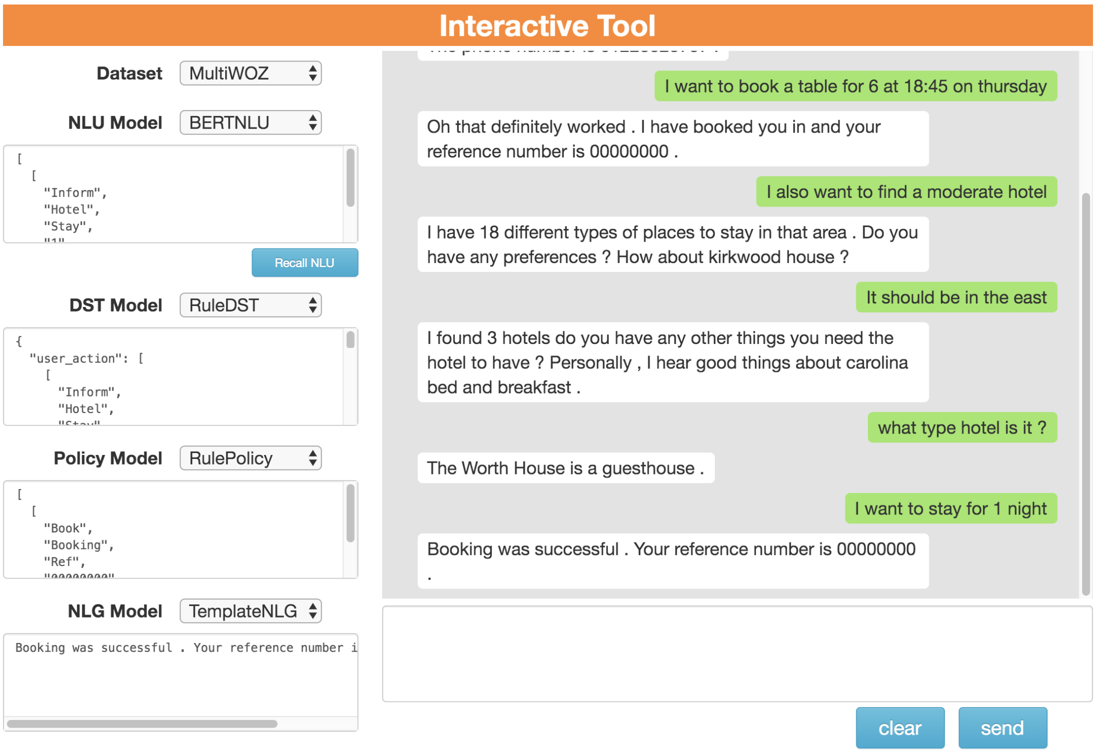
User dialogue acts that cause loop:
- 53% Request-Hotel-Phone-?
- 21% Request-Hotel-Post-?
- 14% Request-Hotel-Addr-?

Table 1: Comprehensive results (partial) of the demo system in Section 3 for the Hotel domain. To save space, only the most frequent errors are presented.

2.4 Analysis Tool

To evaluate a dialogue system quantitatively, ConvLab-2 offers an analysis tool to perform an end-to-end evaluation with a specified user simulator and generate an HTML report which contains rich statistics of simulated dialogues. Charts and tables are used in the test report for better demonstration. Partial results of a demo system in Section 3 are shown in Figure 2 and Table 1. Currently, the report contains the following pieces of information for each task domain:

- Metrics for overall performance such as success rate, inform F1, average turn number, etc.
- Common errors of the NLU component, such as the confusion matrix of user dialogue acts. For example in Table 1, 34% of the requests for the Postcode in the Hotel domain are misinterpreted as the requests in the Hospital domain.
- Frequent invalid, redundant, and missing system dialogue acts predicted by the dialogue policy.
- The system dialogue acts from which the NLG component generates responses that confuse the user simulator. For the example in Table 1, it is hard to inform the user that the hotel has free parking.
- The causes of dialogue loops. Dialogue loop is the situation that the user keeps repeating the same request until the max turn number is reached. This result shows the requests that are hard for the system to handle.



Dataset: MultiWOZ

NLU Model: BERTNLU

```
[
  [
    "Inform",
    "Hotel",
    "Stay",
```

DST Model: RuleDST

```
{
  "user_action": [
    [
      "Inform",
      "Hotel",
```

Policy Model: RulePolicy

```
[
  [
    "Book",
    "Booking",
    "Ref",
```

NLG Model: TemplateNLG

```
Booking was successful . Your reference number i
```

I want to book a table for 6 at 18:45 on thursday

Oh that definitely worked . I have booked you in and your reference number is 00000000 .

I also want to find a moderate hotel

I have 18 different types of places to stay in that area . Do you have any preferences ? How about kirkwood house ?

It should be in the east

I found 3 hotels do you have any other things you need the hotel to have ? Personally , I hear good things about carolina bed and breakfast .

what type hotel is it ?

The Worth House is a guesthouse .

I want to stay for 1 night

Booking was successful . Your reference number is 00000000 .

Figure 3: Interface of the Interactive Tool.

The analysis tool also supports the comparison between different dialogue systems that interact with the same user simulator. The above statistics and comparison results can significantly facilitate error analysis and system improvement. We plan to provide a more comprehensive evaluation in the future.

2.5 Interactive Tool

ConvLab-2 provides an interactive tool that enables researchers to converse with a dialogue system through a graphical user interface and to modify intermediate results to correct a dialogue that goes wrong.

As shown in Figure 3, researchers can customize their dialogue system by selecting the dataset and the model of each component. Then, they can interact with the system via the user interface. During a conversation, the output of each component is displayed on the left side as a JSON formatted string, including the user dialogue acts parsed by the N-LU, the belief state tracked by the DST, the system dialogue acts selected by the policy and the final system response generated by the NLG. By showing both the dialogue history and the component outputs, the researchers can get a good understanding of how their system works.

In addition to the fine-grained system output, the interactive tool also supports intermediate output modification. When a component makes a mistake and the dialogue fails to continue, the researchers can correct the JSON output of that component to redirect the conversation by replacing the original output with the correct one. This function is helpful when the researchers are debugging a specific component.

In consideration of the compatibility across platforms, the interactive tool is deployed as a web service that can be accessed via a web browser. To use self-defined models, the researchers have to edit a configuration file, which defines all available models for each component. The researchers can also add their own models into the configuration file easily.

3 Demo

This section demonstrates how to use ConvLab-2 to build, evaluate, and diagnose a traditional pipeline dialogue system developed on the MultiWOZ dataset.

```
import ... # import necessary modules

# Create models for each component
# Parameters are omitted for simplicity
sys_nlu = BERTNLU(...)
sys_dst = RuleDST(...)
sys_policy = RulePolicy(...)
sys_nlg = TemplateNLG(...)

```

```
# Assemble a pipeline system named "sys"
sys_agent = PipelineAgent(sys_nlu, sys_dst,
     sys_policy, sys_nlg, name="sys")

# Build a user simulator similarly but without DST
user_nlu = BERTNLU(...)
user_policy = RulePolicy(...)
user_nlg = TemplateNLG(...)
user_agent = PipelineAgent(user_nlu, None,
     user_policy, user_nlg, name="user")

# Create an evaluator and a conversation environment
evaluator = MultiWozEvaluator()
sess = BiSession(sys_agent, user_agent, evaluator)

# Start simulation
sess.init_session()
sys_utt = ""
while True:
    sys_utt, user_utt, sess_over, reward = sess.
     next_turn(sys_utt)
    if sess_over:
        break
print(sess.evaluator.task_success())
print(sess.evaluator.inform_F1())

# Use the analysis tool to generate a test report
analyzer = Analyzer(user_agent, dataset="MultiWOZ")
analyzer.comprehensive_analyze(sys_agent,
     total_dialog=1000)
```

Listing 1: Example code for the demo.

To build such a dialogue system, we need to instantiate a model for each component and assemble them into a complete agent. As shown in the above code, the system consists of a BERTNLU, a rule-based DST, a rule-based system policy, and a template-based NLG. Likewise, we can build a user simulator that consists of a BERTNLU, an agenda-based user policy, and a template-based NLG. Thanks to the flexibility of the framework, the DST of the simulator can be None, which means passing the parsed dialogue acts directly to the policy without the belief state.

For end-to-end evaluation, ConvLab-2 provides a BiSession class, which takes a system, a simulator, and an evaluator as inputs. Then this class can be used to simulate dialogues and calculate end-to-end evaluation metrics. For example, the task success rate of the system is 64.2%, and the inform F1 is 67.0% for 1000 simulated dialogues. In addition to automatic evaluation, ConvLab-2 can perform human evaluation via Amazon Mechanical Turk using the same system agent.

Then the analysis tool can be used to perform a comprehensive evaluation. Equipped with a user simulator, the tool can analyze and compare multiple systems. Some results are shown in Figure 2 and Table 1. We collected statistics from 1000 simulated dialogues and found that

- The demo system performs the poorest in the Hotel domain but always completes the goal in the Hospital domain.
- The sub-task in the Hotel domain is more likely to cause dialogue loops than in other domains. More than half of the loops in the Hotel domain are caused by the user request for the phone number.
- One of the most common errors of the NLU component is misinterpreting the domain of user dialogue acts. For example, the user request for the Postcode, address, and phone number in the Hotel domain is often parsed as in other domains.
- In the Hotel domain, the dialogue acts whose slots are Parking are much harder to be perceived than other dialogue acts.

The researchers can further diagnose their system by observing fine-grained output and rescuing a failed dialogue using our provided interactive tool. An example is shown in Figure 3, in which the BertNLU falsely identified the domain as *Restaurant*. After correcting the domain to *Hotel* manually, a *Recall NLU* button appears. By clicking the button, the dialogue system reruns this turn by skipping the NLU module and directly use the corrected NLU output. Combined with the observations from the analysis tool, alleviating the domain confusion problem of the NLU component may significantly improve system performance.

4 Code and Resources

ConvLab-2 is publicly available on https://github.com/thu-coai/ConvLab-2[2]. Resources such as datasets, trained models, and tutorials will also be released. We will keep track of new datasets and state-of-the-art models. Contributions from the community are always welcome.

5 Conclusion

We present ConvLab-2, an open-source toolkit for building, evaluating, and diagnosing a task-oriented dialogue system. Based on ConvLab (Lee et al., 2019b), ConvLab-2 integrates more powerful models, supports more datasets, and develops an analysis tool and an interactive tool for comprehensive end-to-end evaluation. For demonstration, we give an example of using ConvLab-2 to build, evaluate, and diagnose a system on the MultiWOZ dataset. We hope that ConvLab-2 is instrumental in promoting the research on task-oriented dialogue.

[2] The site will become accessible after the legal process is completed.

References

Tom Bocklisch, Joey Faulkner, Nick Pawlowski, and Alan Nichol. 2017. Rasa: Open source language understanding and dialogue management.

Pawel Budzianowski, Iñigo Casanueva, Bo-Hsiang Tseng, and Milica Gasic. 2018a. Towards end-to-end multi-domain dialogue modelling.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018b. MultiWOZ - a large-scale multi-domain wizard-of-Oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026, Brussels, Belgium. Association for Computational Linguistics.

Wenhu Chen, Jianshu Chen, Pengda Qin, Xifeng Yan, and William Yang Wang. 2019. Semantically conditioned dialog response generation via hierarchical disentangled self-attention. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3696–3709, Florence, Italy. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyag Gao, and Dilek Hakkani-Tur. 2019. Multiwoz 2.1: Multi-domain dialogue state corrections and state tracking baselines. *arXiv preprint arXiv:1907.01669*.

Jianfeng Gao, Michel Galley, and Lihong Li. 2019. Neural approaches to conversational ai. *Foundations and Trends® in Information Retrieval*, 13(2-3):127–298.

Izzeddin Gür, Dilek Hakkani-Tür, Gokhan Tür, and Pararth Shah. 2018. User modeling for task oriented dialogues. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 900–906. IEEE.

Matthew Henderson, Blaise Thomson, and Jason D. Williams. 2014. The second dialog state tracking challenge. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 263–272, Philadelphia, PA, U.S.A. Association for Computational Linguistics.

Hwaran Lee, Jinsik Lee, and Tae-Yoon Kim. 2019a. SUMBT: Slot-utterance matching for universal and scalable belief tracking. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5478–5483, Florence, Italy. Association for Computational Linguistics.

Sungjin Lee, Qi Zhu, Ryuichi Takanobu, Zheng Zhang, Yaoqin Zhang, Xiang Li, Jinchao Li, Baolin Peng, Xiujun Li, Minlie Huang, and Jianfeng Gao. 2019b. ConvLab: Multi-domain end-to-end dialog system platform. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 64–69, Florence, Italy. Association for Computational Linguistics.

Wenqiang Lei, Xisen Jin, Min-Yen Kan, Zhaochun Ren, Xiangnan He, and Dawei Yin. 2018. Sequicity: Simplifying task-oriented dialogue systems with single sequence-to-sequence architectures. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1437–1447, Melbourne, Australia. Association for Computational Linguistics.

Mike Lewis, Denis Yarats, Yann Dauphin, Devi Parikh, and Dhruv Batra. 2017. Deal or no deal? end-to-end learning of negotiation dialogues. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2443–2453, Copenhagen, Denmark. Association for Computational Linguistics.

F. Mairesse, M. Gasic, F. Jurcicek, S. Keizer, B. Thomson, K. Yu, and S. Young. 2009. Spoken language understanding from unaligned data using discriminative classification models. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4749–4752.

Alexander Miller, Will Feng, Dhruv Batra, Antoine Bordes, Adam Fisch, Jiasen Lu, Devi Parikh, and Jason Weston. 2017. ParlAI: A dialog research software platform. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 79–84, Copenhagen, Denmark. Association for Computational Linguistics.

Alexandros Papangelis, Mahdi Namazifar, Chandra Khatri, Yi-Chia Wang, Piero Molino, and Gokhan Tur. 2020. Plato dialogue system: A flexible conversational ai research platform.

Baolin Peng, Xiujun Li, Lihong Li, Jianfeng Gao, Asli Celikyilmaz, Sungjin Lee, and Kam-Fai Wong. 2017. Composite task-completion dialogue policy learning via hierarchical deep reinforcement learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2231–2240, Copenhagen, Denmark. Association for Computational Linguistics.

Osman Ramadan, Paweł Budzianowski, and Milica Gašić. 2018. Large-scale multi-domain belief tracking with knowledge sharing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 432–437, Melbourne, Australia. Association for Computational Linguistics.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2019. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. *arXiv preprint arXiv:1909.05855*.

Jost Schatzmann, Blaise Thomson, Karl Weilhammer, Hui Ye, and Steve Young. 2007. Agenda-based user simulation for bootstrapping a POMDP dialogue system. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, pages 149–152, Rochester, New York. Association for Computational Linguistics.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Ryuichi Takanobu, Hanlin Zhu, and Minlie Huang. 2019. Guided dialog policy learning: Reward estimation for multi-domain task-oriented dialog. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 100–110, Hong Kong, China. Association for Computational Linguistics.

Stefan Ultes, Lina M. Rojas-Barahona, Pei-Hao Su, David Vandyke, Dongho Kim, Iñigo Casanueva, Paweł Budzianowski, Nikola Mrkšić, Tsung-Hsien Wen, Milica Gašić, and Steve Young. 2017. PyDial: A multi-domain statistical dialogue system toolkit. In *Proceedings of ACL 2017, System Demonstrations*, pages 73–78, Vancouver, Canada. Association for Computational Linguistics.

Tsung-Hsien Wen, Milica Gašić, Nikola Mrkšić, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically conditioned LSTM-based natural language generation for spoken dialogue systems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1711–1721, Lisbon, Portugal. Association for Computational Linguistics.

Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, Milica Gašić, Lina M. Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2017. A network-based end-to-end trainable task-oriented dialogue system. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 438–449, Valencia, Spain. Association for Computational Linguistics.

Ronald J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3):229–256.

Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. 2019. Transferable multi-domain state generator for task-oriented dialogue systems. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 808–819, Florence, Italy. Association for Computational Linguistics.

Tiancheng Zhao, Kaige Xie, and Maxine Eskenazi. 2019. Rethinking action spaces for reinforcement learning in end-to-end dialog agents with latent variable models. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1208–1218, Minneapolis, Minnesota. Association for Computational Linguistics.

Qi Zhu, Kaili Huang, Zheng Zhang, Minlie Huang, and Xiaoyan Zhu. 2020. CrossWOZ: A large-scale Chinese cross-domain task-oriented dialogue dataset. *Transactions of the Association for Computational Linguistics*.