ConvLab: Multi-Domain End-to-End Dialog System Platform

Sungjin Lee[†] **Qi Zhu**[‡] **Ryuichi Takanobu**[‡] **Zheng Zhang**[‡] **Yaoqin Zhang**[‡] **Xiang Li**[‡]
Jinchao Li[†] **Baolin Peng**[†] **Xiujun Li**[†] **Minlie Huang**[‡] **Jianfeng Gao**[†]
[†]Microsoft Research, USA [‡]Tsinghua University, China
[‡]{zhu-q18,gxly15,z-zhang15,zhangyq17}@mails.tsinghua.edu.cn
[†]{sule,jincli,xiul,jfgao}@microsoft.com [‡]aihuang@tsinghua.edu.cn

Abstract

We present *ConvLab*, an open-source multi-domain end-to-end dialog system platform, aiming to enable researchers to quickly set up experiments with reusable components and compare a large set of different approaches, ranging from conventional pipeline systems to end-to-end neural models, in common environments. ConvLab offers a set of fully annotated datasets and the associated pre-trained reference models. As a showcase, we extend the MultiWOZ dataset with user dialog act annotations to train all component models and demonstrate how ConvLab makes it easy and effortless to conduct complicated experiments in multi-domain end-to-end dialog settings.

1 Introduction

Despite decades of research on dialog and increasingly large amounts of (annotated) dialog datasets, it is still challenging for any team who is new to the area to quickly develop a reasonable baseline system for task-oriented dialog due to the lack of a well-structured, easy-to-use open-source system that allows researchers to build and evaluate dialog bots. ConvLab is aimed to fill the gap. ConvLab is an open-source multi-domain end-to-end dialog system that allows researchers to automatically train dialog models, build and evaluate task-completion dialog bots. Such open-source systems have been instrumental in many AI research breakthroughs. For example, among many, Moses (Koehn and et al., 2007), HTK (Young et al., 2002) and CoreNLP (Manning et al., 2014) have been widely used to facilitate subsequent research in machine translation, speech recognition and natural language processing, respectively.

ConvLab consists of a rich set of modeling tools and runtime engines for building task-oriented bots of different types, and an end-to-end evaluation platform. There are roughly two architectures of dialog systems (Gao et al., 2019): (1) a modular architecture (the top layer in Figure 2), consisting of natural language understanding (NLU), dialog state tracker (DST), dialog policy (POL) and natural language generation (NLG) components; and (2) a fully end-to-end neural architecture (the bottom layer in Figure 2) to minimize laborious hand-coding and error propagation down the pipeline. There also have emerged some models in-between (Wen et al., 2016; Mrkšić et al., 2016). Due to the wide range of approaches and different metrics used in prior studies, it is been impractical to perform a rigorous comparative study under the same condition. ConvLab is the first dialog research platform that covers a full range of trainable statistical models with fully annotated datasets, differing from previous toolkits whose focus is largely concentrated on the system policy component while other components are mostly limited to pre-fixed baseline models (Ultes and et al., 2017; Miller et al., 2017; Li et al., 2018).

There is also an increasing interest in building bots that seamlessly intertwine multiple subdomains to accomplish high-level user goals (Peng et al., 2017; Budzianowski et al., 2018). The development of multi-domain dialog system adds additional complexities to both data collection and annotation, and the models for dialog system components. For the former, Budzianowski et al. (2018) collected the MultiWOZ dataset, a dialog corpus with dialogs ranging over multiple domains for the trip information setting, whereas there is no open-platform yet that is designed to handle multi-domain, multi-intent phenomena. To foster multi-domain dialog research, ConvLab features the MultiWOZ task and offers a complete set of reference models ranging from individual components to end-to-end models that are trained on the MultiWOZ data with additional annotation for user dialog acts which is missing from the original

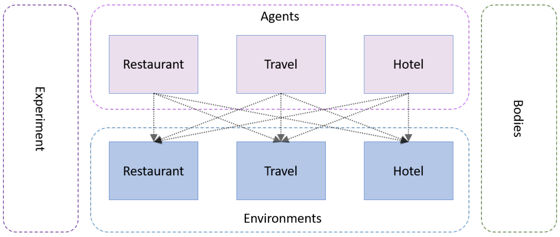

Figure 1: Overall design of ConvLab.

MultiWOZ dataset. Furthermore, ConvLab will be the standard platform for the multi-domain end-to-end task-completion dialog track in DSTC8[1].

Finally, to support end-to-end evaluation, ConvLab offers two complementary modules: Amazon Mechanical Turk integration for human evaluation and simulated users for automated evaluation. For user simulation, ConvLab provides both rule-based simulators and data-driven simulators. As data-driven user simulation recently gains more traction, ConvLab makes another contribution as a research platform for advancing user simulation technologies.

The summary of the unique contributions of ConvLab is:

- To the best of our knowledge, ConvLab is the first open-source multi-domain end-to-end dialog system that covers a full range of trainable statistical models with associated annotated datasets.
- ConvLab provides a rich set of tools and recipes to develop dialog systems of different types, enabling researchers to compare widely different approaches under the same condition.
- ConvLab provides end-to-end evaluation via both human and simulators.
- We are organizing DSTC8 and releasing ConvLab to the public.

2 ConvLab

This section details the design of ConvLab and its flexibility to support a wide range of experiments.

2.1 Overall Design

At a high level, to support flexible architectures for multi-domain dialog, ConvLab embraces the Agents-Environments-Bodies (AEB) design (illustrated in Figure 1) with the following semantics (Wah Loon Keng, 2017):

Agent an instance of dialog agent.

Environment an instance of user simulator or human evaluation component.

Body an incarnation of an agent in the environment – each body stores data that is specific to the associated agent and environment (indicated by the edges in Figure 1): states, actions, rewards, done flags.

With the AEB design, besides the usual single agent and single environment setting, a variety of advanced research experiments, such as multi-agent learning, multi-task learning and role-play, can be conducted without requiring specialized code for each case.

Multi-agent learning A centralized agent maps the joint observation of all domains to a joint action. A major drawback of this approach is its exponential growth in the observation-action space with the number of domains. One can address this intractability by factoring the centralized spaces into multi-agent systems (including hierarchical reinforcement learning agents). For example, in Figure 1, the centralized agent *Travel* can be decomposed into two separate domain agents *Restaurant* and *Hotel*.

Multi-task learning An agent can have multiple bodies in different environments for the sake of transfer learning. For example, any agent in Figure 1 can have its bodies not only in the corresponding environment but also in other environments to learn common knowledge across multiple domains. For example, in Figure 1, each agent can learn from all available environments.

Role play Recently, there has been increased interest in leveraging self-play as an alternative way of training reinforcement learning agents (Silver and et al., 2017). Following the same spirit, for task-completion dialogs, one can devise a role play – one agent plays the role of the system while the other agent the user. Such a role play setting can be readily achieved by having two agents talk to each other though a round-robin environment.

[1] https://sites.google.com/dstc.community/dstc8/home

To perform systematic comparisons of agents and environments, and to automate hyper-parameter search, ConvLab makes use of SLM Lab (Wah Loon Keng, 2017) and Ray[2] for the experiment component in Figure 1 which provides multi-level control layers, i.e. Session, Trial and Experiment, and produces evaluation reports for each layer.

Session Each session initializes the agents and environments and then runs for a pre-defined number of episodes.

Trial Each trial holds a fixed set of parameter values and runs multiple sessions with random seeds. The trial then analyzes the sessions and takes the average.

Experiment An experiment is a study where the hyper-parameters are treated as input variables, and the outcome is measured by task-specific metrics such as success rate and average reward. Search is then automatically conducted to find the hyper-parameters that yield best performance.

ConvLab also helps avoid specifying complicated command line parameters and writing scripts by enabling users to control all relevant functionality via JSON configuration files. A configuration file specifies the model and its parameters for each component of the agent and environment for a given experiment. Thanks to the flexible configuration layer, researchers can build an array of different agents (Section 2.2) and environments (Section 2.3) with only slight modifications in the configuration file. Some example configuration files are presented in Section 4.

2.2 Dialog Agent Configuration

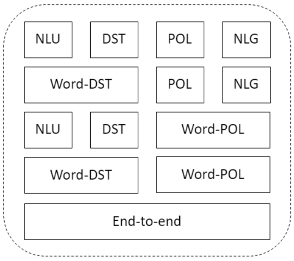

Figure 2: A dialog system configuration view.

In Figure 2, each layer represents a different way of constructing a dialog system. The top layer, for example, corresponds to the conventional pipeline architecture consisting of NLU, DST, POL and NLG. Recently, researchers have introduced some models that merge some of the typical components such as word-level dialog state tracking, word-level dialog policy and end-to-end models, resulting in various possible combinations for building a dialog system as shown from the second layer in Figure 2. However, comparison among these possibilities in an end-to-end setting has been largely overlooked, partly due to the burden of implementing all comparative systems. With ConvLab, researchers can now focus on any particular component in Figure 2 while testing the algorithm in an end-to-end setting by simply creating a configuration file with a specification of other components.

2.3 Environment Configuration

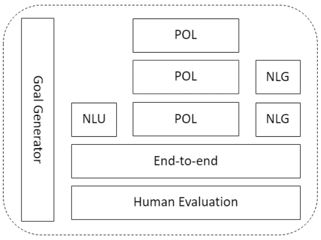

Figure 3: An environment configuration view.

As shown in Figure 3, there are also many different ways of combining components to build an environment. For example, the top layer corresponds to a user simulator operating at the dialog act level which is the typical setting of prior works focusing on developing reinforcement learning methods for dialog policy optimization. As with dialog agent, there are recent attempts on end-to-end approaches to avoid requiring expensive annotation (Kreyssig et al., 2018). For human evaluation, ConvLab also provides an integration of crowd source platform such as Amazon Mechanical Turk[3] as shown in the bottom layer.

2.4 Reference Models

This section describes a set of reference models for each component that are available in the initial release. As we will keep adding new state-of-the-art models, the set of reference models available in ConvLab will be extended.

[2] `https://github.com/ray-project/ray`

[3] ConvLab makes use of ParlAI's MTurk library (`http://parl.ai/static/docs/mturk.html`).

Natural Language Understanding For natural language understanding, ConvLab provides three reference models: Semantic Tuple Classifier (STC) (Mairesse et al., 2009), OneNet (Kim et al., 2017) and Multi-intent LU (MILU). STC can handle multi-domain, multi-intent dialog acts but cannot detect out-of-vocabulary (OOV) values. While OneNet can capture OOVs, it cannot handle multi-intent dialog acts. Thus, ConvLab offers a new MILU model which extends OneNet to process multi-intent dialog acts. For more details on MILU, please refer to the ConvLab site.

Dialog State Tracking The dialog state tracker is responsible for updating the belief state. ConvLab provides a rule-based tracker similar to the baselines in DSTCs (Williams et al., 2013) that are adapted to handle multi-domain interactions.

Word-level Dialog State Tracking Word-level DSTs directly take system and user natural language as inputs and update dialog state. ConvLab imports MDBT (Ramadan et al., 2018) model which jointly identifies the domain and tracks the belief states by utilizing the semantic similarity between dialog utterances and ontology terms.

System Policy For system policy, ConvLab provides three classes of implementations: handcrafted policy, supervised learning policy and reinforcement learning policy. For reinforcement learning, ConvLab supports a set of popular algorithms: DQN (Mnih et al., 2013) and its variants, REINFORCE (Williams, 1992), PPO (Schulman et al., 2017) and its self-imitation variant (Oh et al., 2018) . For multi-domain dialog, ConvLab initially offers centralized policies where the policy maps the joint observation of all domains to a joint action and will feature decentralized multi-agent approaches as well as hierarchical reinforcement learning approaches (Peng et al., 2017).

Natural Language Generation ConvLab provides a template-based model and SC-LSTM (Wen et al., 2015) for natural language generation. Each model is able to take the multi-domain, multi-intent dialog acts as input.

Word-level Policy Following Wen et al. (2016), word-level policy directly maps a context to response. ConvLab imports the baseline implementation released for the benchmarking purpose by Budzianowski et al. (2018)[4]. The baseline model extends a sequence-to-sequence model (Sutskever et al., 2014) with a dialog state encoding and a database query result encoding as additional features to the decoder.

User Policy For user policy, ConvLab provides an agenda-based (Schatzmann et al., 2007) user model and data-driven approaches such as HUS and its variational variants (Gur et al., 2018). Similar to the system side, each model works at the dialog act level, and can be pipelined with NLU and NLG modules to construct a whole user simulator.

End-to-end Model ConvLab makes available two end-to-end dialog system models: Mem2Seq (Madotto et al., 2018) and Sequicity (Lei et al., 2018). To support multi-domain intents, Sequicity resets the belief span when the model predicts a topic shift between domains.

3 Domains

The initial release of ConvLab offers two domains of differing complexity: MultiWOZ and Movie.

MultiWOZ The main task of the MultiWOZ domain is to help a tourist in a various situations involving multiple sub-domains such as requesting basic information about attractions and booking a hotel room. Specifically, there are 7 sub-domains - `Attraction, Hospital, Police, Hotel, Restaurant, Taxi, Train`. The annotated data consists of 10,438 dialogs. The average number of turns are 8.93 and 15.39 for single and multi-domain dialogs, respectively. ConvLab features additional annotations for user dialog acts and pre-trained reference models for all dialog system components and user simulators. Furthermore, ConvLab provides a set of end-to-end neural dialog models that are trained on the data.

Movie ConvLab imports the `Movie` domain from Microsoft Dialog Challenge (Li et al., 2018), encouraging researchers to continue working on the movie ticket booking task with enhanced tools. The annotated dataset consists of 2,890 dialogs, with approximately 7.5 turns per dialog on average. ConvLab offers a complete reference set of

[4] `https://github.com/budzianowski/multiwoz`

models trained on the data for both agent and user simulator.

We plan to add more domains such as the Taxi and Restaurant domains from Microsoft Dialog Challenge.

4 Demo

To demonstrate a glimpse of some working systems, this section presents two end-to-end experiments: 1) a comparison between NLU with rule-based DST and word-level DST; 2) a comparison between rule-based policy with NLG and word-level policy.

Experiment 1 Word-level DSTs often have shown higher performance than typical DSTs that take input from NLU (Ramadan et al., 2018; Mrkšić et al., 2016), but none of prior works confirmed the performance improvement in an end-to-end setting. Thanks to the flexible configuration interface and pre-trained reference models, with ConvLab, one can easily set up end-to-end experiments by simply modifying a few lines in the config files as listed in Table 1. While the overall accuracies of the rule-based DST and the word-level DST are 90.2% and 89.7%, respectively, the end-to-end task success rates are 69.05% and 16.67%. This clearly shows the gap between component-level performance and end-to-end performance. A detailed analysis on this is left for future work.

Experiment 2 Though word-level policy obtains an increasing traction, most studies only report corpus-based metrics such as BLEU and pseudo-success rate (i.e. success means all requested attributes are answered). This makes it hard to compare such approaches with conventional policies that are typically evaluated with task success metrics. Due to the space limitations, we omit the experimental config which is largely the same as the config listed on the left column in Table 1 except that the policy and nlg sections under the agent section are now replaced with a corresponding word_policy section. While the reported pseudo-success rate on test data is 60.96%, the success rate with a user simulator is 16.16%. This is also much lower than 69.05%, the performance of its counterpart with rule-based policy and NLG. Thus, there is huge room for improvement of the word-level policy in end-to-end settings.

NLU and rule-based DST

```
{"multiwoz": {
  "agent": [{
   "name": "DialogAgent",
    "nlu": {
     "name": "OneNet"
    },
   "dst": {
    "name": "RuleDST"
   },
   "policy": {
    "name": "ExternalPolicy",
    "algorithm": {
     "name": "RulePolicy"
    }
   },
   "nlg": {
    "name": "TemplateNLG",
    "is_user": false
   }
  }],
  "env": [{
   "name": "multiwoz",
   "nlu": {
    "name": "OneNet"
   },
   "policy": {
    "name": "UserPolicyAgenda"
   },
   "nlg": {
    "name": "TemplateNLG",
    "is_user": true
   }
   "max_t": 40,
   "max_tick": 20000,
  }],
  "body": {
   "product": "outer",
   "num": 1
}}}
```

Word-level DST

```
{"multiwoz": {
  "agent": [{
   "name": "DialogAgent",
   "word-dst": {
    "name": "MDBT"
   },
   "policy": {
    "name": "ExternalPolicy",
    "algorithm": {
     "name": "RulePolicy"
    }
   },
   "nlg": {
    "name": "TemplateNLG",
    "is_user": false
   }
  }],
  "env": [{
   "name": "multiwoz",
   "nlu": {
    "name": "OneNet"
   },
   "policy": {
    "name": "UserPolicyAgenda"
   },
   "nlg": {
    "name": "TemplateNLG",
    "is_user": true
   }
   "max_t": 40,
   "max_tick": 20000,
  }],
  "body": {
   "product": "outer",
   "num": 1
}}}
```

Table 1: Example configs for comparing a system using word-level DST (right) with one using NLU and rule-based DST (left).

5 Code and Resources

The ConvLab platform is publicly available from http://convlab.github.io. Datasets and other resources such as tutorials and documentations can be found in the site.

6 Conclusion

We presented *ConvLab*, an open-source multi-domain end-to-end dialog system platform, that enables researchers to quickly set up experiments and compare different approaches without much effort. We will keep extending ConvLab by adding new state-of-the-art models going forward. The multi-domain end-to-end task completion dialog track in DSTC8 will employ ConvLab as the challenge platform, giving rise to a reference use case.

References

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. Multiwoz-a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. *arXiv preprint arXiv:1810.00278.*

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 Gur, Dilek Hakkani-Tur, Gokhan Tur, and Pararth Shah. 2018. User modeling for task oriented dialogues. *arXiv preprint arXiv:1811.04369*.

Young-Bum Kim, Sungjin Lee, and Karl Stratos. 2017. Onenet: Joint domain, intent, slot prediction for spoken language understanding. In *ASRU*, pages 547–553. IEEE.

Philipp Koehn and et al. 2007. Moses: Open source toolkit for statistical machine translation. In *ACL*, pages 177–180.

Florian Kreyssig, Inigo Casanueva, Pawel Budzianowski, and Milica Gasic. 2018. Neural user simulation for corpus-based policy optimisation for spoken dialogue systems. *arXiv preprint arXiv:1805.06966*.

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 *ACL*, pages 1437–1447.

Xiujun Li, Sarah Panda, Jingjing Liu, and Jianfeng Gao. 2018. Microsoft dialogue challenge: Building end-to-end task-completion dialogue systems. *arXiv preprint arXiv:1807.11125*.

Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. 2018. Mem2seq: Effectively incorporating knowledge bases into end-to-end task-oriented dialog systems. *arXiv preprint arXiv:1804.08217*.

François Mairesse, Milica Gasic, Filip Jurcicek, Simon Keizer, Blaise Thomson, Kai Yu, and Steve Young. 2009. Spoken language understanding from unaligned data using discriminative classification models. In *ICASPP*, pages 4749–4752. IEEE.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *ACL*, pages 55–60.

Alexander H Miller, Will Feng, Adam Fisch, Jiasen Lu, Dhruv Batra, Antoine Bordes, Devi Parikh, and Jason Weston. 2017. Parlai: A dialog research software platform. *arXiv preprint arXiv:1705.06476*.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. 2013. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

Nikola Mrkšić, Diarmuid O Séaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve Young. 2016. Neural belief tracker: Data-driven dialogue state tracking. *arXiv preprint arXiv:1606.03777*.

Junhyuk Oh, Yijie Guo, Satinder Singh, and Honglak Lee. 2018. Self-imitation learning. *arXiv preprint arXiv:1806.05635*.

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. *arXiv preprint arXiv:1704.03084*.

Osman Ramadan, Paweł Budzianowski, and Milica Gašić. 2018. Large-scale multi-domain belief tracking with knowledge sharing. *arXiv preprint arXiv:1807.06517*.

Jost Schatzmann, Blaise Thomson, Karl Weilhammer, Hui Ye, and Steve Young. 2007. Agenda-based user simulation for bootstrapping a pomdp dialogue system. In *NAACL*, pages 149–152.

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

David Silver and et al. 2017. Mastering the game of go without human knowledge. *Nature*, 550(7676):354.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *NIPS*, pages 3104–3112.

Stefan Ultes and et al. 2017. Pydial: A multi-domain statistical dialogue system toolkit. *ACL, System Demonstrations*, pages 73–78.

Laura Graesser Wah Loon Keng. 2017. Slm-lab. `https://github.com/kengz/SLM-Lab`.

Tsung-Hsien Wen, Milica Gasic, Nikola Mrkšić, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. In *EMNLP*, pages 1711–1721.

Tsung-Hsien Wen, David Vandyke, Nikola Mrksic, Milica Gasic, Lina M Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2016. A network-based end-to-end trainable task-oriented dialogue system. *arXiv preprint arXiv:1604.04562*.

Jason Williams, Antoine Raux, Deepak Ramachandran, and Alan Black. 2013. The dialog state tracking challenge. In *SIGDIAL*, pages 404–413.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256.

Steve Young, Gunnar Evermann, Mark Gales, Thomas Hain, Dan Kershaw, Xunying Liu, Gareth Moore, Julian Odell, Dave Ollason, Dan Povey, et al. 2002. The htk book. *Cambridge university engineering department*, 3:175.