metaCAT: A Metadata-based Task-oriented Chatbot Annotation Tool

Ximing Liu, Wei Xue, Qi Su, Weiran Nie, Wei Peng [*]
Huawei Technologies Co., Ltd., PRC
{liuximing1,xuewei5,qi.su,nieweiran,peng.wei1}@huawei.com

Abstract

Creating high-quality annotated dialogue corpora is challenging. It is essential to develop practical annotation tools to support humans in this time-consuming and error-prone task. We present metaCAT, which is an open-source web-based annotation tool designed specifically for developing task-oriented dialogue data. To the best of our knowledge, metaCAT is the first annotation tool that provides comprehensive metadata annotation coverage to the domain, intent, and span information. The data annotation quality is enhanced by a real-time annotation constraint-checking mechanism. An Automatic Speech Recognition (ASR) function is implemented to allow users to paraphrase and create more diversified annotated utterances. metaCAT is publicly available for the community. [1]

1 Introduction

The progress of the development of multi-turn task-oriented dialogue systems has been largely constrained by the availability of large-scale high-quality data (Zhu et al., 2020). Due to the complexity of annotating task-oriented dialogue data, a large number of errors have been found in existing benchmark datasets, e.g., MultiWOZ (Budzianowski et al., 2018). Despite ongoing efforts such as re-annotating the state tags based on the original utterances in MultiWOZ 2.1 (Eric et al., 2019) and introducing slot span annotations in MultiWOZ 2.2 (Zang et al., 2020), there is still much room for improvement. Creating high-quality annotated dialogue corpora is highly challenging thus necessitates a high level of human engagements. It is essential to develop practical annotation tools for supporting experts in this time-consuming and error-prone task.

Although many NLP annotation tools exist, LIDA (Collins et al., 2019) is the only one designed specifically for annotating multi-turn task-oriented benchmark dialogue datasets like MultiWOZ. As a lightweight interactive dialogue annotator, LIDA provides an end-to-end pipeline for converting raw text to structured conversation data. It claims to support machine learning-based annotation recommenders and provide an interface resolving inter-annotators disagreements. LIDA is an easy-to-use tool for tagging Boolean and String type slot values. However, it lacks functionality for annotating complex data. These data include enumeration, span information, and multi-valued slot values for utterance and other metadata such as domains, intents, etc.

A critical step in ensuring data quality when annotating is to perform real-time constraint-checking. Retrospective inter-annotator disagreement resolution (i.e., after an annotation task is completed) is an option offered by LIDA. It is desirable to design a function which provides real-time constraint-checking to eliminate human errors. For example, the slot value "dontcare" should be avoided in the system's annotation, because it can only be applied to a user's annotations. This highlights the need to introduce metadata to define the scope of the annotation and, in the meantime, enforce consistency between human inputs and underlying constraints.

Data annotation task is time-consuming as there are large amounts of typing actions involved. To improve the efficiency of the annotation task, we identify the need to integrate the Automatic Speech Recognition (ASR) function to the annotation process. By automatically converting speech to text, ASR can speed up the data input process otherwise done by keyboard typing. ASR can also be used to create paraphrases of utterances, and as a consequence, enhance the diversity of the spoken

[*]Corresponding author
[1]https://github.com/lexmen318/metaCAT

language. This is especially useful in scenarios where manually collected dialogue corpora are rare and expensive. Providing such diversity is regarded as the key to increasing the robustness of dialogue models trained with these data.

In this paper, we propose metaCAT, which is a web-based task-oriented chatbot annotation tool with complete management of users, tasks, datasets, and dialogues. metaCAT extends LIDA by contributing additional key useful features including:

- comprehensive metadata annotation coverage to the domain, intent and span information w.r.t. each dialogue turn;
- real-time annotation constraint-checking to ensure data quality;
- ASR paraphrasing to speed up an annotator's data input process and increase the diversity of utterances.

2 Related Work

Various annotation tools have been developed, targeting NLP tasks in recent years. As depicted in Collins et al. (2019), these tools are built for disparate NLP tasks ranging from general text processing (i.e., GATE (Cunningham et al., 2002)) to entity linking (e.g., INCEpTION (Klie et al., 2018)). Among them, a number of dialogue annotation tools are discussed below.

DialogueView (Yang and Heeman, 2005) is a dialogue annotation tool developed for segmenting recorded dialogue into utterances, annotating speech repairs, tagging speech acts, and segmenting dialogue into hierarchical discourse segments. The focus is on dialogue segmentation; therefore features for generating task-oriented dialogue data, for example slot-value labeling, are not available.

TWIST (Pluss, 2012) supports turn segmentation and content feature annotation. After highlighting and creating new turn segments, users can assign predefined content features to the targeted turn segment. The content feature includes descriptions like "Objective", "Subjective", etc. However, there is no means to support label classification and slot-value annotation.

DART (Weisser, 2016) stands for Dialogue Annotation and Research Tool, which is a research environment enabling users to annotate and analyze dialogues automatically. DART was developed for linguistic research and analysis. DART provides a convenient means for leveraging linguistic resources to improve the annotation results and test hypotheses. Only intent and enumeration slot can be labeled via this tool. It is not designed to create task-oriented dialogue data for training dialogue models, therefore lacking most of the related features.

LIDA (Collins et al., 2019) claims to be the first annotator capable of providing a full end-to-end dialogue annotation pipeline. It is a web-based tool designed specifically for task-oriented dialogue systems. It provides basic features to annotate the Boolean or String type slot-value of each turn of the dialogue. LIDA integrates with back-end machine learning models as annotation recommenders to achieve semi-automatic annotating. It contains a dedicated interface to resolve inconsistent annotations between different annotators. Despite implementing several key features for annotating task-oriented dialogue data, LIDA lacks functionality supporting users to handle complex data annotation, such as enumeration, span information and multi-valued slot labeling. LIDA can only handle inter-annotator disagreement resolution at the post-editing stage. Real-time annotation error-checking is out of the scope.

In this paper, we present metaCAT, which is designed to fill the above gaps. Table 1 provides a detailed comparison of the features and capabilities provided by metaCAT and other dialogue annotation tools.

3 System Overview

metaCAT consists of a front-end and a back-end. The back-end is implemented with Flask providing the RESTful service for data manipulation while the front-end is developed with the VUE.js framework. metaCAT uses MongoDB as a database.

The user interfaces for the administrator and annotators are developed independently as the roles and privileges are different. The administrator is responsible for uploading metadata and dialogue data, assigning and monitoring the annotation tasks, and downloading annotated data. Annotators are responsible for the execution of annotating tasks and paraphrasing tasks. All tasks are managed in batches with each containing several to dozens of dialogues. The intermediate results of annotation are saved to MongoDB.

metaCAT defines a comprehensive annotation ontology system for building task-oriented dia-

Features	metaCAT	LIDA	TWIST	DART
Classification Labels	YES	YES	NO	YES
Edit Dialogues/Turns	YES	YES	YES	YES
Turn/Dialogue Segmentation	YES	YES	YES	YES
Recommenders	YES	YES	NO	NO
Annotation Metadata Import	YES	NO	NO	NO
Comprehensive Tagging	YES	Boolean/String Slot	NO	Intent/Enum. Slot
Paraphrasing (via ASR or Keyboard)	YES	NO	NO	NO
Real-time Constraint-Checking	YES	NO	NO	NO
Comprehensive Management	YES	Dialogue Only	NO	Dialogue Only
Multilingual Support	YES	NO	NO	NO

Table 1: A detailed comparison of the features offered by metaCAT and other dialogue annotation tools. "Comprehensive Tagging" indicates if a tool provides annotation for a comprehensive range of items including Utterance/Domain/Intent/Slot/Span. "Comprehensive Management" refers to the management of a comprehensive range of items including User/Task/Dataset/Dialogue. "Enum." stands for enumeration.

logue data. We follow the way how SGD (Rastogi et al., 2019) organizes metadata:

- **General Domain**: The general domain contains intents with no slots, for example, "Thank" or "Bye";
- **Service Domain**: These domains involve specific intents with slots. An utterance usually belongs to one service domain, but in some rare cases it may contain two or more service domains, such as an utterance booking "Hotel" and "Taxi" at the same time;
- **Intent**: An utterance usually implies one or more specific intents.
- **Slot**: Slots generally represent key information carried with the intent.

The slots can be classified into enumeration type and string type. An enumeration slot contains a list of specific slot values without any span information. A string slot is usually one word or short segment of text taken from the utterance. Some special values of a string slot exist without span information, e.g., "dontcare". A slot usually corresponds to one value but there are some slots containing multiple values, such as names of two restaurants recommended by the system.

In addition to offering a comprehensive metadata annotation ontology, metaCAT also supports a dialogue paraphrasing feature, which requires the annotators to paraphrase the utterances via keyboard or ASR before the annotation task. Figure 1 depicts the journey of the administrator and annotators in using metaCAT. A short video clip is available in our YouTube channel [2] demonstrating the system.

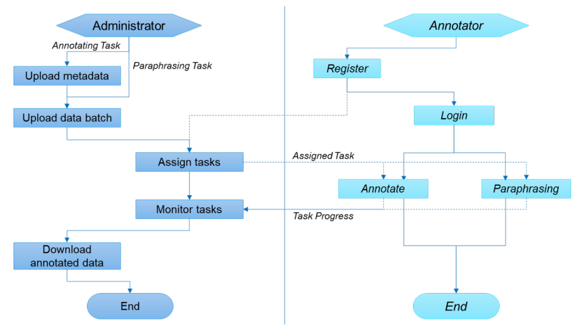


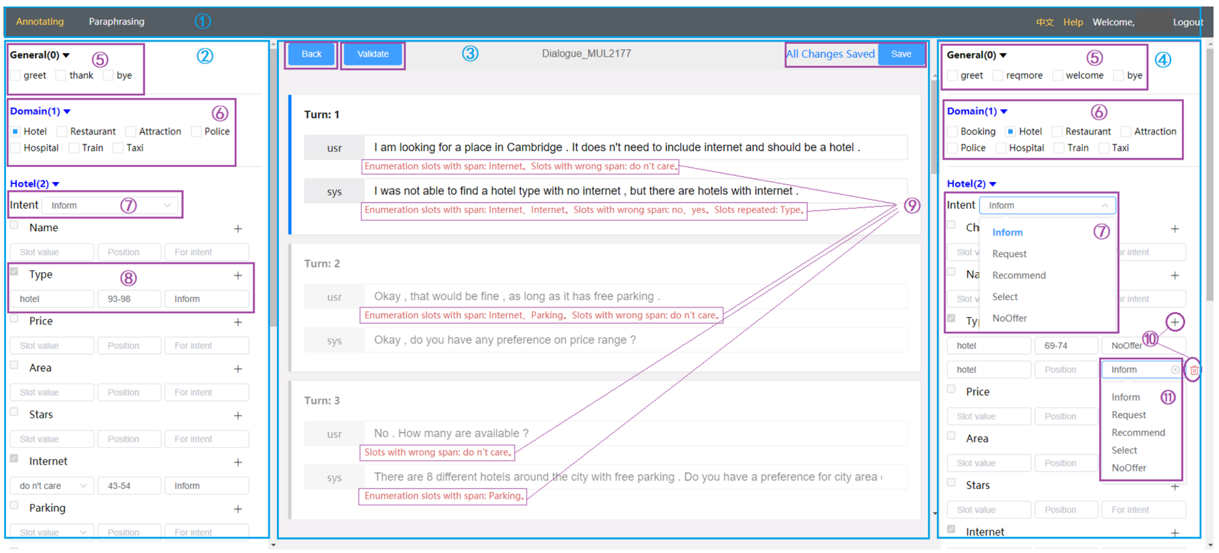
Figure 1: The journey of the administrator and annotators using metaCAT.

3.1 Administrator Journey

Before creating annotation tasks, the administrator needs to upload metadata of the dataset, which is generated offline based on a provided template file in JSON format. After the metadata is uploaded, dialogue data for annotating can be imported. Currently, metaCAT supports the following formats of dialogue data:

- **Raw Format**: It contains crowd-sourced raw utterances from both the user side and the system side without any annotation.
- **Annotating Format**: This is the same format as the output file of metaCAT. It includes both utterances and associated annotations. This is the format used for annotating in metaCAT, which may have been converted from formats used in other annotating tools.

[2]https://youtu.be/07_PWD4_c4E


Annotating
Paraphrasing
①
中文 Help Welcome, Logout
General(0) ⑤ ②
greet thank bye
Domain(1) ⑥
Hotel Restaurant Attraction Police Hospital Train Taxi
Hotel(2)
Intent Inform ⑦
Name
Slot value Position For intent
Type ⑧
hotel 93-98 Inform
Price
Area
Stars
Internet
do n't care 43-54 Inform
Parking
Back Validate ③ Dialogue_MUL2177 All Changes Saved Save
⑨
General(0) ⑤ ④
greet reqmore welcome bye
Domain(1) ⑥
Booking Hotel Restaurant Attraction Police Hospital Train Taxi
Hotel(2)
Intent Inform ⑦
Inform Request Recommend Select NoOffer
⑩
hotel 69-74 NoOffer
hotel Position Inform
Inform ⑪ Request Recommend Select NoOffer
Price
Area
Stars
Internet

Turn: 1

usr I am looking for a place in Cambridge . It does n't need to include internet and should be a hotel .

Enumeration slots with span: Internet。Slots with wrong span: do n't care。

sys I was not able to find a hotel type with no internet , but there are hotels with internet .

Enumeration slots with span: Internet、Internet。Slots with wrong span: no、yes。Slots repeated: Type。

Turn: 2

usr Okay , that would be fine , as long as it has free parking .

Enumeration slots with span: Internet、Parking。Slots with wrong span: do n't care。

sys Okay , do you have any preference on price range ?

Turn: 3

usr No . How many are available ?

Slots with wrong span: do n't care。

sys There are 8 different hotels around the city with free parking . Do you have a preference for city area

Enumeration slots with span: Parking。

Figure 2: Annotating Interface - to annotate "usr" and "sys" utterance. Note: ①:menu bar; ②: "usr" annotating zone; ③ utterance zone; ④: "sys" annotating zone; ⑤: general intent selection; ⑥: service domain selection; ⑦: default intent selection; ⑧:slot-value editing box; ⑨:validating results; ⑩:slot-value adding/deletion; ⑪:slot attached intent modification.

- **Specific Format**: Only one specific format is supported currently, which is MultiWOZ format. It is automatically converted to metaCAT format after uploaded.

- **Paraphrasing Format**: It applies to the paraphrased utterances and related annotations. Data format conversion is performed off-line before paraphrasing.

The result data can be exported in three types of format: annotating format, MultiWOZ format, and paraphrasing format.

3.2 Annotator Journey

After registration, annotators can be assigned to annotating and/or paraphrasing tasks. By clicking the assigned tasks, an annotator can access associated dialogues via the main annotating and paraphrasing interface.

3.2.1 Dialogue Annotating

As shown in Figure 2, the annotating interface is activated by a user's selection of the menu bar (①). The annotating interface consists of a left "usr" annotating zone (②), a middle utterance zone (③) and a right "sys" annotating zone (④). The following steps are performed to annotate a dialogue turn:

- **Utterance Editing**: Manual editing may be involved to fix typos or ASR-induced errors.

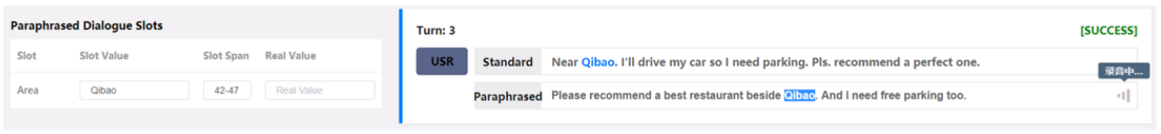
- **Domain and Intent Selection**: Both left and right annotating zones provide annotators with functions to select a general intent (⑤) of the current utterance, e.g., "Thank" or "Bye". The service domain (⑥) and all related intents (⑦) for the current utterance can be assigned in this zone.

- **Slot Annotating**: There are two ways to annotate a slot. A user can select a span of the utterance text (i.e., a restaurant name) and drag it to the slot-value editing box (⑧). The other way is to click the slot-value editing box and select one value from the opened drop-down list box. For an enumeration slot, only the second method is supported. For a string slot, the first method is used by default and whether the second method is available depends on the specific slot-values defined in the metadata.

During annotating, metaCAT enables automatical constraint-checking by validating the annotated results against predefined constraints (shown as ⑨ in Figure 2). The constraint-checking is activated under the circumstances depicted as below:

- **Turn Switching**: After an annotator finishes one dialogue turn, the validation is performed.

- **Validating or Saving**: When the "Validate" or "Save" button is clicked, metaCAT performs validating for all turns of dialogues.



Paraphrased Dialogue Slots

Slot	Slot Value	Slot Span	Real Value
Area	Qibao	42-47	Real Value

Turn: 3 [SUCCESS]

USR Standard: Near Qibao. I'll drive my car so I need parking. Pls. recommend a perfect one.

Paraphrased: Please recommend a best restaurant beside Qibao. And I need free parking too.

Figure 3: A screenshot of the paraphrasing interface of metaCAT.

Task		#Dialogue	#Turn/Dialogue	#Turn	#Non-span-slot	#Span-slot	#Slot
LIDA	Expert	79	3.5	276.5	816.5	0	816.5
	Novice	60	3.5	210	617	0	617
metaCAT	Expert	25	15.56	389	315.5	561.3	876.8
	Novice	20	15.56	311.2	252.4	449	701.4

Table 2: Hourly annotated dialogues in the re-annotation of 100 dialogues sampled from MultiWOZ 2.1 using metaCAT. "#" stands for "The number of".

Task	#Dialogue	#Turn	#Slot
ASR Exclusive	16.7	344.8	485.3
ASR Inclusive	28.6	612.4	863.7

Table 3: A comparison of paraphrasing and re-annotation of a bespoke cross-domain dialogue dataset with and without ASR during eight working hours. "#" stands for "The number of".

The validating process is designed to detect common annotating errors, e.g., repeated slots, repeated slot-values, overlapped spans, and ill-matched intents. The constraints are predefined or enforced by the metadata definition.

3.2.2 Dialogue Paraphrasing

The paraphrasing interface (Figure 3) is similar to the annotating interface with some minor differences. It is divided into two areas with the slot area on the left-hand side and the utterance area on the right side. The entire annotating operation includes the following steps:

- **Utterance Paraphrasing**: The annotator paraphrases the utterance using the colloquial language keeping the basic semantics, intent and slot unchanged.
- **Slot Annotating**: The annotator selects a span of the utterance and drags it to the slot-value editing box.
- **Real-value Filling**: For slots with real values, the annotator needs to fill the values to the corresponding slots after paraphrasing.

The validating process is basically the same as that of the annotating task except for the underlying detecting rules. There is no need to detect ill-matched intents, but the paraphrased utterance's integrity and diversity are checked quantitatively using Levenshtein distance.

4 Evaluation

metaCAT is used to correct annotation errors in MultiWOZ 2.1. One hundred dialogues are sampled from the test set and imported into metaCAT for evaluation. The benchmark data are those published in LIDA (Collins et al., 2019) on a different sample. Four annotators have been employed in this task, two of whom were novice users and the others are experienced annotators.

Table 2 captures the results of annotating a sample of MultiWOZ 2.1 using metaCAT on an hourly basis. Expert metaCAT users produce an average of 876.8 slot annotations, while novice users generate 701.4 annotations on average. Note that we cannot directly compare the results with those produced in LIDA (Collins et al., 2019) as different data are included in the experiment. However, the hourly annotation rate of metaCAT for both novice and expert users indicates an equivalent efficiency to those of LIDA. Table 3 shows that the inclusion of ASR increases the hourly annotation rate of paraphrasing and re-annotation by 1.8 folds.

LIDA is not able to handle metadata therefore dynamic domain switching is not available. Contrastively, It is easy to switch domain annotating as metaCAT clearly defines metadata and captures constraints among the domain, intent, slot and slot-value. In addition, drag-and-drop is an efficient way to handle span information annotation.

5 Conclusion and Future Work

We present metaCAT, an open-source web-based annotation tool designed specifically for task-oriented dialogue datasets. It is the first annotation tool providing a comprehensive metadata annotation to cover service domain, intent and span information. metaCAT ensures data quality with a real-time constraint-checking mechanism. An ASR-inclusive paraphrasing function enables users to generate more diversified utterances with annotations. metaCAT can be used to enhance task-oriented dialogue datasets (i.e., MultiWOZ 2.1) with easy-to-use functionality.

Existing dialogue annotation tools focus on the current turn of the dialogue, lacking inclusion of related contexts. For instance, there is a need to refer to the information from the previous turns to annotate the term "the hotel" in a typical hotel booking dialogue scenario. Apart from the above-mentioned co-referencing annotation function, future work will focus on designing other annotation features to meet the community's practical needs, for example, annotations for sentiment and multi-modal intent. Many open-domain conversations are goal-driven with the intent to solve real-world problems, for example, for recommendation and search purposes. Developing knowledge-grounded open-domain conversation datasets becomes an emerging trend (Gopalakrishnan et al., 2019) to regulate the free-form data with facts and knowledge. metaCAT can be customized to annotate open-domain dialogue datasets from this perspective.

Acknowledgments

We are grateful to colleagues from Huawei Technologies Co., Ltd. for insightful discussions and technical supports.

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. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026, Brussels, Belgium. Association for Computational Linguistics.

Edward Collins, Nikolai Rozanov, and Bingbing Zhang. 2019. LIDA: Lightweight interactive dialogue annotator. 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): System Demonstrations*, pages 121–126, Hong Kong, China. Association for Computational Linguistics.

Hamish Cunningham, Diana Maynard, Kalina Bontcheva, and Valentin Tablan. 2002. GATE: an architecture for development of robust HLT applications. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 168–175, Philadelphia, Pennsylvania, USA. 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*.

Karthik Gopalakrishnan, Behnam Hedayatnia, Qinlang Chen, Anna Gottardi, Sanjeev Kwatra, Anu Venkatesh, Raefer Gabriel, and Dilek Hakkani-Tr. 2019. Topical-Chat: Towards Knowledge-Grounded Open-Domain Conversations. In *Proc. Interspeech 2019*, pages 1891–1895.

Jan-Christoph Klie, Michael Bugert, Beto Boullosa, Richard Eckart de Castilho, and Iryna Gurevych. 2018. The INCEpTION platform: Machine-assisted and knowledge-oriented interactive annotation. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 5–9, Santa Fe, New Mexico. Association for Computational Linguistics.

Brian Pluss. 2012. *TWIST Dialogue Annotation Tool*.

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

Martin Weisser. 2016. Dart the dialogue annotation and research tool. *Corpus Linguistics and Linguistic Theory*, 12(2):355 – 388.

Fan Yang and Peter A. Heeman. 2005. DialogueView: an annotation tool for dialogue. In *Proceedings of HLT/EMNLP 2005 Interactive Demonstrations*, pages 20–21, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Xiaoxue Zang, Abhinav Rastogi, and Jindong Chen. 2020. MultiWOZ 2.2 : A dialogue dataset with additional annotation corrections and state tracking baselines. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 109–117, Online. Association for Computational Linguistics.

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