A first study on a dialogue system for an academic writing aid system

Data adaption from restaurant reservation to writing aid and measurement of accuracy of dialogue act prediction and slot filling

Ni Zihui

Graduate School of Information, Production and Systems

Waseda University

A thesis submitted for the degree of

Master of Engineering

July 8th, 2021

Thesis Supervisor: Professor Yves Lepage

Abstract

Our main goal is to build a dialogue system for an academic writing aid system. We thus need to build a new dataset for the writing aid task and to finetune a dialogue model for this task. Current task-based dialogue systems have many domains, such as restaurant reservation, ticket booking and take-away ordering. Writing aid tasks are missing in such datasets. We automatically modify sentences and annotations in existing dialogue datasets to adapt to our model and finetune on this new writing aid task with an existing dialogue frame on the adapted data. Our evaluation is the accuracy of dialogue act prediction and slot filling. We design a semantic frame for the writing aid task, and design slots and corresponding slot values for semantic frames. Since each dialogue sentence has its corresponding slot and slot value and the generation of answers depends on the prediction of slots and slot values, we take the prediction accuracy of dialogue act and slot value as the final evaluation criterion. We modified DSTC 2 data for the restaurant reservation task to make it a data set suited for the writing aid task. On this new dataset, we use the DeepPavlov dialogue framework to train and evaluate. The accuracy of the dialogue act in our writing aid task is 51% and the precision, recall and F1 score of the slot filling could reach 98%, 93% and 96%. Our dialogue system can help users in answering questions about handwritten paragraphs and modifying paragraphs to a certain extent, so as to complete the functions of writing aid.

Keywords: pipeline dialogue system, writing aid, task-oriented

Acknowledgements

I would like to express my gratitude to all those who helped me during the writing of this thesis. Thanks to my two years in Waseda University, I learned the qualities that a researcher should have. Thanks to Professor Lepage and the students in the laboratory for their guidance, I have benefited a lot from them. Thank my friends around me for taking care of my study and life. In the past two years, I have made clear my goals, enriched my knowledge and made some progress. This has a good impact on my future work research.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 The dialogue system

A dialogue system (Eckert et al. (1997)) is a computer system intended to converse with a human. Task-oriented dialogue systems (Yan et al. (2017)), compared to chatbot, are dialogue systems that help users complete certain tasks and usually need to query on the external knowledge base and return the correct results for meeting the specific needs of the user. Pipeline-based task-oriented dialogue system (Liu et al. (2017)) consists of natural language understanding, dialogue management, and natural language generation modules. The current task-based dialogue system has been applied to many fields, such as hotel reservations, ticket booking, and food reservations. However, in the field of writing aid (Chang et al. (2008)), there is no complete dialogue system that can meet the needs of non-native speakers' academic writing modification. The writing aid dialogue dataset also does not exist. For popular fields, there is a lot of dialogue data corresponding to the task for the existing dialogue system to train. However, the writing aid field is not a popular field. So even if we can build a dialogue system, there is no data for training.

Figure 1.1. shows how the dialogue system satisfies the request of the user. In the example, we can see that the user input the paragraph to the system and wants a modification about the paragraph. To make the paragraph better, the system ask the user whether he or she needs a grammar check service. In the grammar check task, the writing aid system could just modify the paragraph or give an error report to the user. So the dialogue system asks the user if he or she just want an error report or just modify the paragraph directly. What the dialogue system does is answer the question of the user and return the new paragraph modified by the writing aid system.

The paragraph written by user: *With a natural language as a means of communication with a computer system, the users can make a question or a statement in the way they normally think about the information being discussed, freeing them from having to know how the computer stores or processes the information.*

Robot: Hello, welcome to the writing aid system. You can do many kinds of modification of the paragraph here. Do you want a grammar check first?
User: Yes, please.
Robot: Okay, do you want an error report or just modify the paragraph?
User: An error report please.

The paragraph modified by the writing aid system: *With ~~a~~ natural language as a means of communication with a computer system, the users can make a question or a statement in the way they normally think about the information being discussed, freeing them from having to know how the computer stores or processes the information.*
The error report: "With a natural ...": "a" should be deleted in the sentence.

Figure 1.1: An example of the dialogue between user and the system, the paragraph simulating user writing is from Bird et al. (2008)

The existing task-oriented dialogue systems are divided into two categories: end-to-end dialogue system and pipeline dialogue system. The pipeline dialogue system, as shown in Figure 1.2, divides the dialogue characters into five parts for implementation and evaluation. The five parts are:

- Audio signal recognition: This step is to collect users' needs, which can be text input or audio input. If it is the input of audio, we need to recognize the audio as text first, and then carry on the semantic analysis of the text.

- Natural language understanding (NLU): In this part, we need to judge the user's input and fill in the present slot value to extract the words related to the demand from the user's words. And through a certain method of classification, find the category of requirements and fill in the known information given by users.

- Dialogue state tracker (DST): Tracking dialogue states is the core component to ensure a robust manner in dialog systems. It estimates the user's goal at every turn of the dialogue. We need such a module to ensure that all the information required by the current demand has been input by the user. For example, if the user wants a cup of coffee, the user needs to input the coffee type, temperature and other information. If the information is true, it needs to be detected and asked to the user.

- Knowledge base (KB): The dialogue management module sometimes needs to interact with the external knowledge base. Firstly, the external knowledge of unstructured text related to conversation context is selected by TF-IDF, and then the knowledge is represented by RNN. Finally, context and external knowledge are taken into account when the matching score of candidate reply is calculated.Then the generation operation is carried out: an open domain dialogue generation system based on the answer of knowledge base is established by memory network. The core idea is to find the knowledge in the knowledge base in the dialogue context, transform the knowledge into semantic vector representation, integrate it with the dialogue context vector representation, and input it into the decoder to generate the reply

- Natural language generation (NLG): The NLG process in Task-based dialogue is based on NLU (domain classification and intention identification, slot filling) and DST, and generates dialogue reply according to the learned strategies. The general reply includes clarifying requirements, guiding users, inquiring, confirming, and concluding remarks.

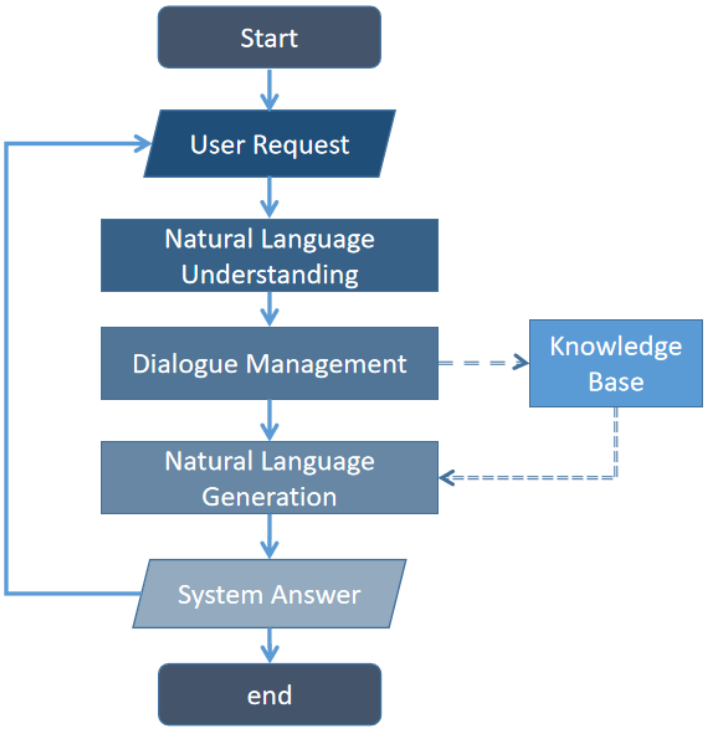

Figure 1.2: This is the flow chart of traditional pipeline dialogue system. When the user sends the request to the pipeline dialogue system, the NLU part needs to change the order sentence into the semantic frame, and then the DM part use the semantic frame to generate the dialogue state (If what the user needs is satisfied), The NLG part uses the dialogue state to generate an answer to the user. (ask for more information or output the automatic answer.)

```
{"speaker": 2,
"text": "Hello, welcome to the Cambridge restaurant
   system. You can ask for restaurants by area, price
   range or food type. How may I help you?",
"dialog_acts": [
{"act": "welcomemsg",
"slots": []
}]}
```

```
{"speaker": 1,
"text": "cheap restaurant",
"dialog_acts": [
{"act": "inform",
"slots": [["pricerange",
           "cheap"]]
}]}
```

Figure 1.3: The example of the DSTC 2 dataset (The data is copied from Henderson et al. (2014a))

Our goals can be divided into two categories:

- **To build a dialogue dataset for the writing aid task.**
  Try to automatically modify the sentences and the annotations in the existing dialogue dataset.

- **To finetune a dialogue model for the writing aid task.**
  In the existing Dialogue framework, finetuning the task with the modified data.

(1) To build a dialogue dataset for the writing aid task. The dataset we built is based on the DSTC 2 dataset. Figure 3.1 shows an example of the DSTC 2 dataset (Henderson et al. (2014a)). The DSTC 2 dataset is a dataset human-machine dialogue data, a multi-round dialogue data set in the field of restaurant reservations, in which there are 1, 612 training data, 506 validation set, and 1, 117 test set. Table 1.1 shows the annotations of the DSTC 2 dataset.

(2) To finetune a dialogue model for the writing aid task.

annotation	explaination
speaker	the system or the user
text	the dialogue text
goal	information about the goal/task for the system
slots	request parameter (slot list)
acts	the purpose of the dialogue sentence, such as inform, inquiry, etc
dialogue_acts	this is determined by what slots and acts is, directly determines how to generate answers and call the APIs

Table 1.1: The annotations of the DSTC 2 dataset

1.2 Slots and semantic frame

1.2.1 Slot

Slot is parameters of the query of user. For instance, when a user queries the task of keyword extraction, the number of keywords is required for the dialogue system to retrieve the appropriate information.Slot filling refers to the process of completing the information in order to make the user's intention into the user's clear instructions. For example, in the dialogue task of booking air tickets, if the user's instruction is "help me book a plane ticket and start from Tokyo." Here, two slots should be filled, with "air ticket" and "Tokyo" as slot values and filled in the slots named "vehicle" and "destination". At this time, in the dialogue system, the system can understand that the user wants to buy a ticket to Hangzhou. Due to the lack of slot value of the slot named "place of departure", there may be a round of inquiry dialogue. If the user answers "Fukuoka", fill "Fukuoka" in the slot named "place of departure" Finally, according to the supplementary information, the system generates instructions and calls the interface to purchase the ticket from Fukuoka to Tokyo.

We usually use the IOB method to do the slot filling. The output sentence is labelled by IOB labeling method, which includes the IOB label. B is the beginning of the slot, I is the inside part of the slot , O is the outside part of the slot. Table 1.3 shows an example of the slot filling.

1.2.2 Semantic frame

Semantic frame in the Natural Language Understanding (NLU) part obtains the state of universe and the the statistic of the paragraph. In Natural Language Understanding part, the dialogue system will fill the slots of the dialogue text according to the semantic frame, so as to understand the meaning

slot	informable	values
food	yes	91 possible values
name	yes	113 possible values
pricerange	yes	3 possible values
addr	no	-
phone	no	-
postcode	no	-
signature	no	-

Table 1.2: The slot table designed in DSTC 2 Henderson et al. (2014a)

Input	I	want	to	apply	keyword	extraction	with	5	words	on	this	paragraph
Output	O	O	O	O	O	O	O	B-number	O	O	O	O

Table 1.3: An example of the IOB labeling method in slot filling

of the user's input.

Table 1.2 shows the semantic frame in the DSTC 2 dataset. Figure 1.2.2 shows the relationship between slot and slot value.

We need to modify the data set from dialogue sentences and dialogue annotations.

1.3 The dialogue system applicated to the writing aid system

In this thesis, we realized a dialogue system dedicated to complete academic writing aid tasks (Huebner (2015)). As it is shown in Figure 1.5, in our dialogue system for writing aid tasks, in order to meet the user's needs of paragraph modification, we need to add the process of inputting paragraph and paragraph modification to the traditional dialogue system. Our main goal is to combine the traditional dialogue system with the writing aid tasks.

Figure 1.6 is the UML graph of our proposed Dialogue System. In this system, the user needs to send the order to the dialogue system and write the paragraph nonacademic, and the writing aid system needs to realize the modification of the paragraph ordered by the user. The dialogue system is the connection of the user and the dialogue system

1.3.1 The tasks of the writing aid system

Our goal in the writing aid system is to change the nonacademic part in the paragragh to be academic. The paragragh written by non-native English

```
{"food":{
 "caribbean":[
 "carraibean",
 "carribean",
 "caribbean"
 ],
 "kosher":[
 "kosher"
 ],
 "tuscan":[
 "tuscan"
 ],
 "french":[
 "french"
 ]}}
```

```
{"this":{
 "dontcare":[
 "dont care",
 "doesnt matter",
 "any fine"
 "any noise"
 "any of town"
 "noise anything"
 "any type"
 "anything"
 "any thing"
 "what ever"
 "does not matter"
]}}
```

Figure 1.4: An example of the slot and the corresponding slot value in the DSTC 2 dataset (The graph is copied from Henderson et al. (2014a))

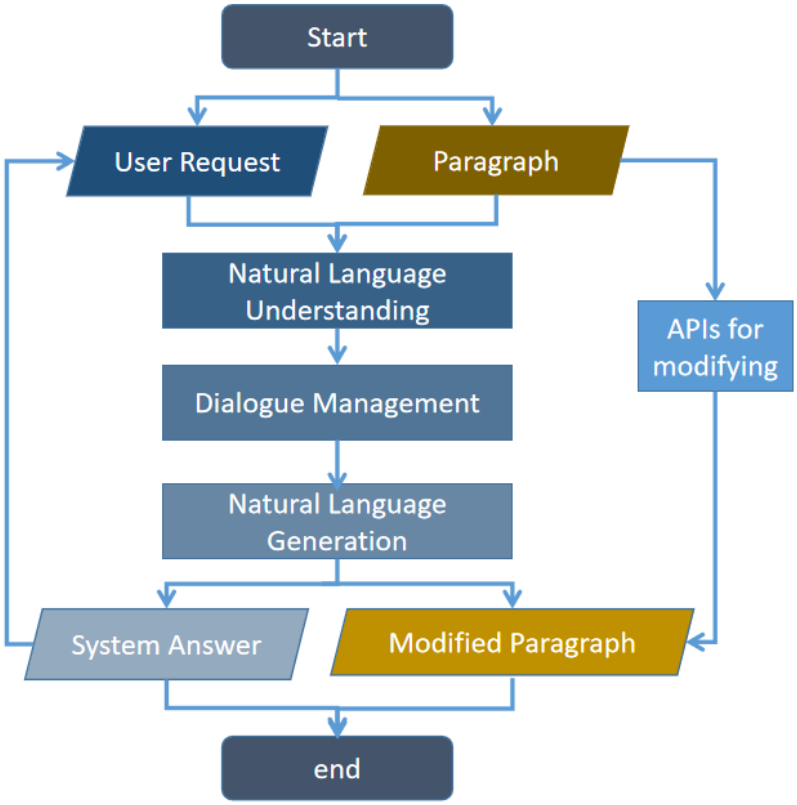

Figure 1.5: The flow chart of the pipeline dialogue system for writing aid tasks

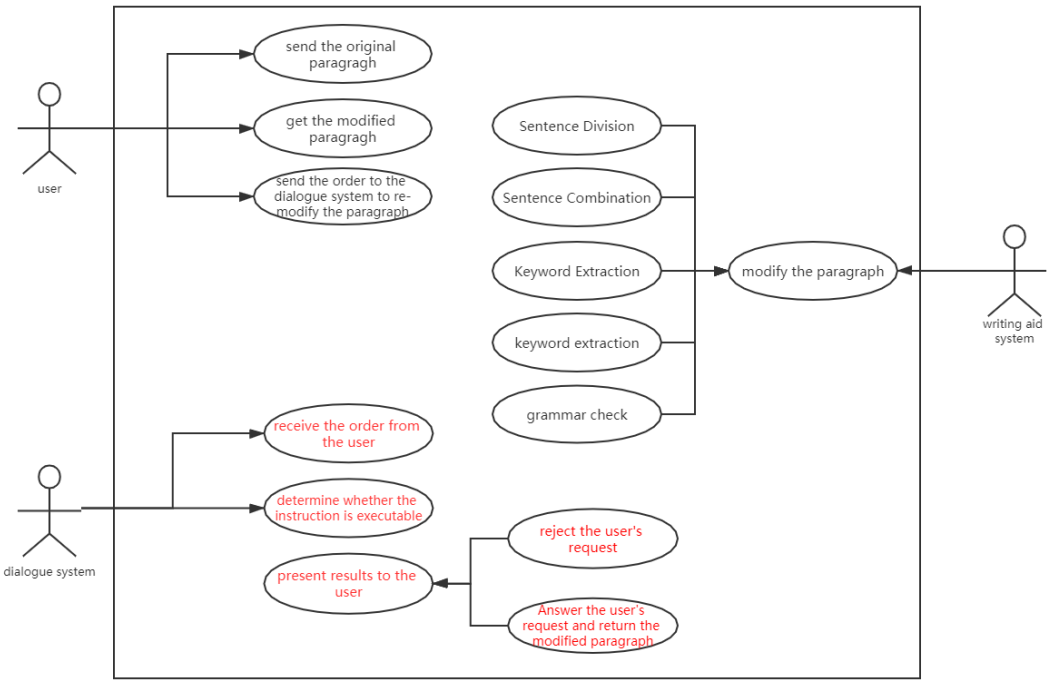

Figure 1.6: The UML graph of our proposed dialogue system.

speaker may have lots of nonacademic part: grammatical error, too long sentence and so on.

Figure 1.7 shows the structure and the writing aid tasks in the dialogue system. There are 4 tasks in this system: sentence division, keyword extraction, grammatical error correction and text summarization.

1.3.2 The frame of the dialogue system

We choose DeepPavlov framework (Burtsev et al. (2018a)) as the framework of our dialogue system. DeepPavlov framework is an open-source conversational AI library built on TensorFlow, Keras and PyTorch.DeepPavlov framework is designed for:

- Development of production ready chat-bots and complex conversational systems
- Research in the area of NLP and, particularly, of dialog systems

1.4 Organization of the thesis

In order to make it more convenient for non-native English speakers to write academic English, in this paper, we mainly propose a method to modify the existing dialogue dataset and implement the dialogue system. In the next chapter, we want to introduce the related work we used. And in Chapter 3, we want to introduce the methods and architecture we proposed. Chapter 4 describes the experiments and the results, and we want to introduce our future work in Chapter 5.

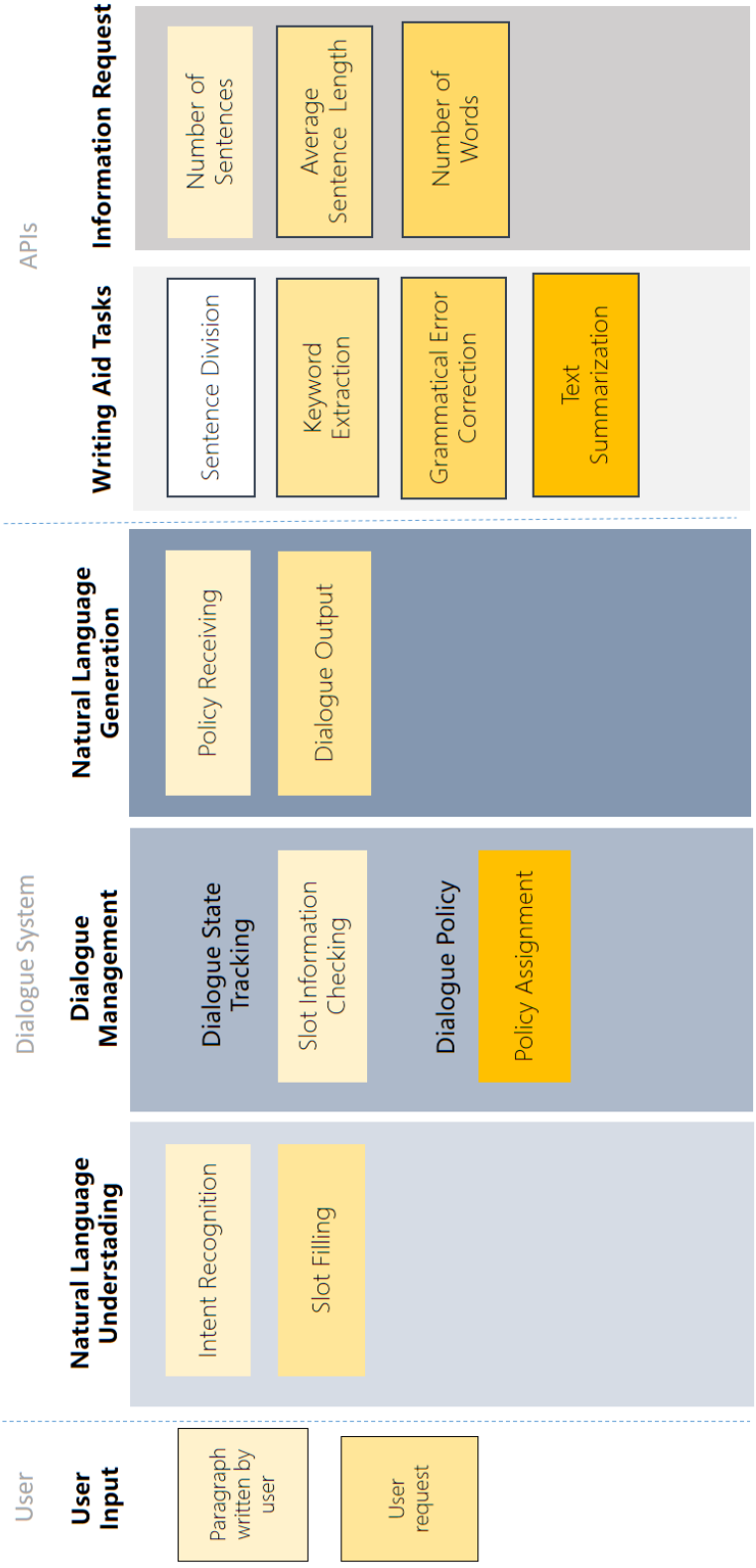

Figure 1.7: The structure of the writing aid dialogue system

Chapter 2

Related Work

2.1 Writing aid system

With regard to writing aid system, (Stasaski et al. (2020)) proposed a novel asynchronous method for collecting CIMA collection tutoring dialogue dataset that is both amenable to the needs of deep learning algorithms and reflective of pedagogical concerns. (Malchanau et al. (2018)) described a method to create dialogue corpora annotated with interoperable semantic information. (Li et al. (2020)) proposed task-based dialogue system provides a simulation environment with goal-oriented characteristics, allowing learners to continuously improve their language fluency in terms of speaking ability by simulating conversational situational exercises. (Malchanau et al. (2018)) built a multitasking Interactive Cognitive Tutor is evaluated as showing human-like proactive and adaptive behaviour in setting goals, choosing appropriate strategies and monitoring processes across contexts, and encouraging the user exhibit similar metacognitive competences.

Name of the dataset	**Size**	**Source**	**Quality**	**Fields**
DailyDialog (Li et al. (2017))	13,118	textbooks	auto-extracted	various topics
Wizard-of-Wikipedia (Dinan et al. (2019))	22,311	crowdsourcing	human-written	various topics
Document-grounded (Feng et al. (2020))	4,470	crowdsourcing	human-written	various topics
Persona-Chat (Zhang et al. (2018))	10,981	crowdsourcing	human-written	various topics
Self-dialogue (Fainberg et al. (2018))	24,165	crowdsourcing	human-written	various topics
Cornell Movie Corpus (Danescu-Niculescu-Mizil and Lee (2011))	304,713	movie scripts	auto-extracted	movie
Self-feeding chatbot (Heinroth et al. (2010))	145,873	human-bot dialogues	human-written (half)	various topics
Twitter corpus (Cieliebak et al. (2017))	9,783	twitter posts/replies	auto-extracted	various topics
Opensubtitles (Lison and Tiedemann (2016))	5,000,000	movie subtitles	auto-extracted	movie
AirDialogue (Wei et al. (2018a))	301,427	human annotators	human-written	travel, flight
MultiWOZ (Budzianowski et al. (2018a))	8,438	human behaviors	human-written	hotel, restaurant
DSTC 2 (Henderson et al. (2014a))	2235	human behaviors	Amazon Mechanical Turk	food

Table 2.1: Some existing datasets of the dialogue system

2.2 Dataset

Task-oriented dialogue systems try to build a conversation model that can accomplish tasks through interactions with humans using natural language processing. (Wei et al. (2018b)) purportedly contains 400,000 templated dialogues about airline reservations, where rule-based environments have also been recently explored to aid the evaluation of goal-oriented dialogue agents. (Budzianowski et al. (2018b)) recently introduced a multi-domain dialogue dataset (MultiWOZ), which adds new challenges in DST due to its mixed-domain conversations and has utilized a variety of data collection techniques. The Dialog State Tracking Challenge (DSTC) proposed by (Henderson et al. (2014b)) was initiated to address this barrier by providing a common testbed and evaluation framework for dialog state tracking algorithms. DSTC 2 have produced bootstrapped human-machine dataset for a restaurant search and become one of the central research topics in the dialogue community. As it shows in Table 2.1 the current task-based dialogue system has been applied to many fields, such as hotel reservations,ticket booking, and take-way ordering. There is no dialogue dataset for the writing aid field. Even if we have a dialogue system, there is no data for training. So we need to build a dialogue dataset dedicated to writing aid dialogue system.

Name of the system	**Dialogue scalability**	**Natural language management**	**Natural language Understanding**	**Natural language generation**
OpenDial (Lison and Kennington (2016a))	direct coding	plan	machine learning	template
RASA (Nguyen and Shcherbakov (2021))	direct coding	machine learning	machine learning	-
ParlAI (Miller et al. (2017a))	direct coding	machine learning	machine learning	template
DeepPavlov (Burtsev et al. (2018a))	direct coding	machine learning	machine learning	template
OwlSpeak (Heinroth et al. (2010))	universal plug and play	information state	rule	template
PyDial (Ultes et al. (2017))	direct coding	reinforcement learning	rule and machine learning	rule and machine learning
DialogStudio (Jung et al. (2008))	direct coding	example	machine learning	example
RavenClaw (Bohus and Rudnicky (2009))	direct coding	plan	rule	template

Table 2.2: Some existing dialogue systems

2.3 Dialogue system

OpenDial Lison and Kennington (2016b), an open-source dialogue system framework based on an information-state architecture, allows system developers to easily specify and edit dialogue behaviours, which is a crucial requirement for commercial conversational applications. Rasa Core, proposed by (Nguyen and Shcherbakov (2021)), is a commercial open-source tool designed for developers that combine frame-based state updates with learning by example dialogue policy. Parlai (Miller et al. (2017b)) is a toolkit for deep learning research in dialog, allowing the simulator to be easily used with deep learning frameworks. DeepPavlov (Burtsev et al. (2018b)) is a deep learning library with a focus on task-oriented dialogue. It provides demos and pre-trained models for tasks such as question answering and sentiment classification. ConvLab-2 (Zhu et al. (2020)) has been made to allow the use of data-driven approaches based on machine learning or machine teaching for the development of dialogue modules. As it shows in Table 2.2, the current task-based dialogue system has been applied to many fields, such as hotel reservations,ticket booking, and take-way ordering. So we need to modify and fine tune a dialogue system dedicated to complete academic writing aid tasks.

Chapter 3

Proposed writing aid dialogue dataset and system

We propose a new dialogue system for writing aid tasks and a method for creating a new writing aid dialogue dataset.

3.1 A dialogue dataset oriented to writing aid tasks

We have five writing aid tasks in our prototype writing aid system, which include *Sentence Division*, *Sentence Combination*, *Keyword Extraction*, *Grammatical Error Correction* and *Text Summarization*. We propose to build the dialogue dataset corresponding with the writing aid tasks. Our dataset is mainly modified from DSTC 2 dataset. The method can be divided into five steps:

- Design the new slot and slot value in the slot table.
- Make the new slot and value appear in the new dialogue sentence.
- Prepare writing aid-related sentence data set.
- Replace the all the words about restaurant reservation in DSTC 2 dataset
- Use grammar check to correct the grammatical errors in the dialogue text.

(1) Replace the slot and value of the slot in the original DSTC 2 dataset in the newly designed slot table.
The first step of data set building is to ensure that the semantic frame of data set is designed for writing aid task. In this way, in the process of natural language understanding, the dialogue system can identify the components in the dialogue sentences to different writing aid tasks. In our proposed method, we replace the slot table of DSTC 2 dataset one-to-one to ensure that all slots and slot values are related to the writing aid task. As the new semantic frame shown in table 4.1, slot represents options different tasks, including writing aid task and natural information of the paragraph. We hope that users can not only put forward the requirements of modifying tasks in the system, but also decide whether to modify by asking the natural information of the paragraph. "Informative" in Table indicates whether the slot value corresponding to this slot is enumerable, and "yes" if it is enumerable. For example, in the grammar check task, the only possible values are "error report" and "modified paragraph", so they are enumerable. If the slot is "number of sentences", the number of sentences in the paragraph can't be enumerated, so the value of "informative" is "no".

slot	informable	numbers of values	values
type of sentence division terms	yes	3	and, but, so
type of grammar error correction	yes	2	error report, modified paragraph
number of keyword extraction	yes	5	1,2,3,4,5
type of text summarization	yes	2	opening sentence, abstract
number of sentences	no	-	-
average number of words	no	-	-
number of total words	no	-	-

Table 3.1: The newly designed slot table

Please demonstrate it with the result with conjunction of word "and".
I want to see the number of sentences.
I want to see the number of total words.
One possible weakness of discourse-based summarization techniques is that they rely greatly on the accuracy of the discourse parser they use .

Table 3.2: The example of the writing aid-related sentence data set

(2) Make the new slot and value appear in the new dialogue sentence.
In this step, we replace the annotation in DSTC 2 dataset and follow the semantic frame table generated in the previous step. After replace all annotations in the JSON file, all the annotations in the original dataset have been replaced by the annotations in our task. Figure 3.1 shows the process of this part.

(3) Prepare writing aid-related sentence data set.
In the previous step, we got the data of replacing words about food with words about writing aid according to the semantic frame table. But there are also some words such as "restaurant" and "eat", which are related to food but do not appear in the semantic frame table. These words should also be replaced. We hope to train a large-scale writing corpus, and then screen these "important" words in the field of food reservation. The selected words form masks in the sentence. Finally, by using the trained model to predict these masks, we can get sentences that are smooth and related to the writing aid tasks.
We get 84 conversations about writing aid manually and added 2,494 sentences in ACL-ARC data set (Bird et al. (2008)). Table 3.2 shows the example sentences of the writing aid-related sentence data set we prepared. [1] as the data set about writing aid.

(4) Replace the all the words about restaurant reservation in DSTC 2 dataset.
In this step, we select the "important" words, use the corpus training model from the previous part, and replace the "important" words. We use pretrained embedding BERT model (Devlin et al. (2019)) to replace the extracted words of DSTC 2 to the words in the writing aid dataset. For each word W_i (Li et al. (2018)) in DSTC 2, we calculate the probability $P_1(w_i)$ in

[1] use the ACL-ARC sentences extracted in the Mango system :/files/Data/AcademicWritingAid/backup/cleaned_data

```
{"speaker": 2,
 "text": "What kind of food would you like?",
 "dialog_acts": [{"act": "request_food ",
 "slots": []}]}
{"speaker": 1,
 "text": "gastropub ",
 "goals": {"food" : "gastropub" },
 "dialog_acts": [{"slots": [["food", "gastropub "]],
 "act": "inform"}]}
```

```
{"speaker": 2,
 "text": "What kind of food →type of grammar error
    correction would you like?",
 "dialog_acts": [{"act": "request_food →
    type_of_grammar_error_correction ", "slots": []}]}
{"speaker": 1,
 "text": "gastropub →error report",
 "goals": {"food"→ type of grammar error correction : "
    gastropub" →error report},
 "dialog_acts": [{"slots": [["food →type of grammar
    error correction ", "gastropub →error report "]], "
    act": "inform"}]}
```

```
{"speaker": 2,
 "text": "What kind of type of grammar error correction
    would you like?",
 "dialog_acts": [{"act": "
    request_type_of_grammar_error_correction ", "slots":
    []}]}
{"speaker": 1,
 "text": " error report",
 "goals": {"type of grammar error correction : " error
    report},
 "dialog_acts": [{"slots": [[" type of grammar error
    correction ", " error report "]], "act": "inform"}]}}
```

Figure 3.1: The example of the DSTC 2 dataset (Data copied from Henderson et al. (2014a))

word	result
restauarant	9.714
korean	9.714
baskey	9.714
lcoated	9.714
Graffiti	9.714
tasca	9.714

Table 3.3: The words are copied from the output of the dataset. The type errors in "restauarant" and "lcoated" come from the data.

DSTC 2 and $P_2(w_i)$ in writing aid dataset. And then we calculate

$$W_i = |log(|P_1(w_i) - P_2(w_i)|)|$$

If W_i of word i is near 0, it should be substituted. The closer W_i is to 0, the greater the frequency difference between the two texts, and the more this word should be replaced.
We use BERT model (Devlin et al. (2019)) to replace extracted words of DSTC 2 to the words in the writing aid dataset. We set the threshold at 9.72. Figure 3.1 shows the part of word that we sift out and need to be replaced.

(5) Use grammar check to correct the grammatical errors in the dialogue text.

Finally, we extract the dialogue sentences in the generated JSON file, and modify the syntax through the syntax modification tool, while others remain unchanged. This step is to ensure that the dialogue sentences in our dataset are smooth.

3.2 The dialogue system

Figure 3.2 shows the process of the task-oriented dialogue system (gobot model) in the DeepPavlov framework. In this model, we use the dialogue actions predicted in the process of dialogue state tracking as the evaluation criteria. In the example shown in Figure 3.2, the predicted result of the model is the most likely dialogue action. During the test, we compare the predicted dialogue action with the dialogue action in the tag. If the predicted dialogue action is the same, the predicted result is correct. In this evaluation process, the more accurate the prediction result of dialogue action is, the higher the ability of the dialogue system to judge the current dialogue behavior is.

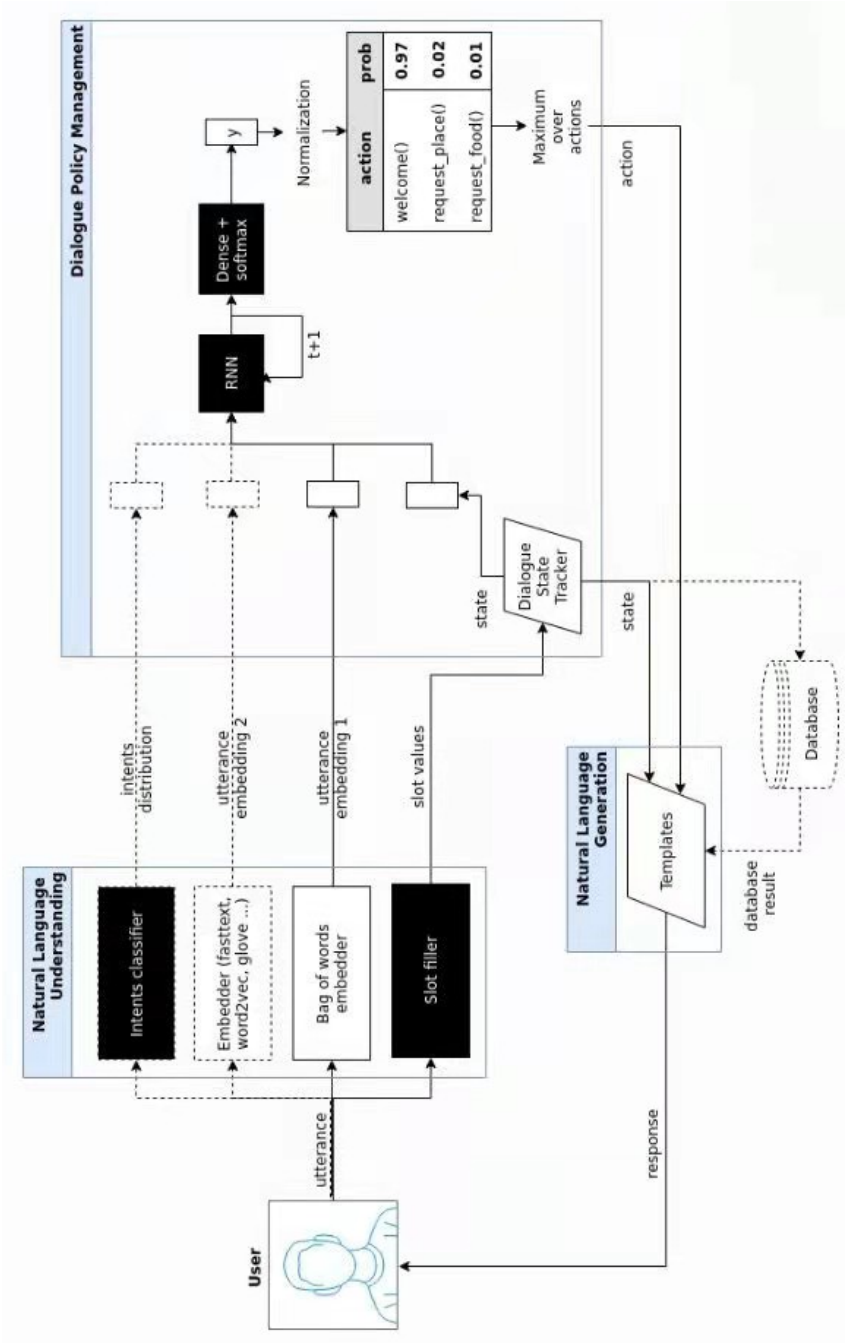

Figure 3.2: The structure of DeepPavlov framework (Graph copied from (Burtsev et al. (2018a)))

Figure 3.2 shows the parameters of gobot model in DeepPavlov framework.

```
{
"model": {
"name": "go_bot",
"network": {
 "name": "go_bot_rnn"
 },
 "slot_filler": {
 "name": "dstc_slotfilling",
 "ner_network": {
 "name": "ner_tagging_network",
  }
  },
 "intent_classifier": {
 "name": "intent_model",
 "embedder": {
  "name": "fasttext"
  },
  "tokenizer": {
  "name": "nltk_tokenizer"
  }
 },
 "embedder": {
 "name": "fasttext"
 },
 "bow_encoder": {
 "name": "bow"
 },
 "tokenizer": {
 "name": "spacy_tokenizer"
 },
 "tracker": {
 "name": "featurized_tracker"
 }
 }
 }
```

Figure 3.3: Parameters of gobot model in DeepPavlov framework

Chapter 4

Experiments and Results

4.1 Dataset

We generate 2,235 sets of dialogue data, containing 39,420 dialogue sentences with their annotations. Table 4.1 shows the examples of the generate data.

4.2 Baselines

We use the gobot model under the DeepPavlov framework to calculate the accuracy of dialogue action and the precision, recall and F1 value of slot filling (SF) task. We use the results of DSTC 2 (Henderson et al. (2014a)), MultiWOZ (Budzianowski et al. (2018a)) and AirDialogue (Wei et al. (2018a)) in these tasks as the baselines.

4.3 Results

We take same numbers with the DSTC 2 dataset (Henderson et al. (2014a)), with 1,612 training data, 506 validation set, and 1,117 test set. Table 4.3 shows the results. From the results, we have obtained quantitative data similar to other datasets, which shows that our data set is effective.

{'speaker': 2, 'text': 'Hello, welcome to the writing aid system. You can do many kinds of modification of the paragraph here. Do you want a grammar check first?', 'dialog_acts': {'act': 'welcomemsg', 'slots': []}}
{'speaker': 1, 'text': 'Yes, please give me the grammar check.', 'goals': {'grammar check': 'dontcare'}, 'dialog_acts': {'slots': ['type of text summarization', 'abstract'], 'act': 'inform'}}
{'speaker': 2, 'text': 'What kind of type of grammar error correction would you like?', 'dialog_acts': {'act': 'request_type_of_grammar_error_correction', 'slots': []}}
{'speaker': 1, 'text': 'any', 'goals': {'type of grammar error correction': 'dontcare', 'type of text summarization': 'dontcare'}, 'dialog_acts': {'slots': [['this', 'dontcare']], 'act': 'inform'}

Table 4.1: The example of our building writing aid dialogue data set

Dataset	**accuracy (DST)**	**precision (SF)**	**recall (SF)**	**F1 (SF)**
DSTC 2	47%	99%	97%	98%
MultiWOZ	47%	96%	96%	96%
AirDialogue	22%	59%	56%	55%
Ours	**51%**	**98%**	**93%**	**0.96%**

Table 4.2: The quantitative result of predicting bot answers on datasets.

Chapter 5

Conclusion and Future Work

In this research, we built a writing aid dialogue dataset based on DSTC 2 and finetune our writing aid dialogue task on the DeepPavlov framework. We obtained high results on dialogue state tracking accuracy and F1 score in the slot filling task.

As a follow up, we hope to test our data set under different dialogue system frameworks. We will try more dialog system tasks to ensure the availability of our dataset. We want to analyze the different results obtained to analyze the advantages and disadvantages of each dialogue system framework in the writing aid tasks. While ensuring the availability of data sets, we also want to try more writing aid tasks, hoping to solve more writing problems for users.

Appendix A

Declaration by the student

I hereby confirm that:

- this thesis, submitted in partial fulfilment for the degree of Master (Engineering) at the Graduate School of Information, Production and Systems, Waseda University, is my original work;
- quotations and citations have been duly identified by use of quotation marks, change in formatting and use of bibliographical references;
- I have upheld the principles of academic integrity, and I certify that:
  - there is no data falsification in this thesis,
  - there is no data fabrication in this thesis,
  - there is no plagiarism in this thesis;
- this thesis has not been submitted previously or concurrently and, will not be submitted by myself in the future, for any other degree at any other institution.

Name: ..... Student N^o: .....

Signature: ..... Date: .....

Appendix B

Program List

The following is a brief documentation of the programs and scripts we created or modified for the work reported in this thesis.

As of September 2021, these programs and scripts can be found on the local server of the EBMT/NLP lab at the following location:

```
http://133.9.48.111:8082/Ni_Zihui/dialogue_systems_
writing_aid
```

In general, they can be categorized into 4 groups:

- Manually modify the data set based on the correspondence
- Filter high frequency words
- Use language models to replace high-frequency words
- Run the dialogue system

B.1 Manually modify the dataset based on the correspondence

The programs development in this section is used in the experiments reported in Chapter 3. We manually modify the DSTC 2 data set based on the corresponding relationship of the slot. This involves complex modifications to the json file.

B.1.1 change_slot.py

Name	change_slot.py
Type	Python 3.7 script
Version	1
Function	Change the text and slot of the DSTC 2 dataset
Argument 1	The file path of original DSTC 2 dataset
Output	The modified dataset with DSTC 2 format
Location	http://133.9.48.111:8082/Ni_Zihui/dialogue_systems_writing_aid

Table B.1: Description of change_slot.py program

B.1.2 Input

The input of change_slot.py is the standard format of DSTC 2:

```
1 {"speaker": 2,
2 "text": "Hello, welcome to the Cambridge restaurant
    system. You can ask for restaurants by area, price
    range or food type. How may I help you?",
3 "dialog_acts": [{"act": "welcomemsg", "slots": []}]}
4
5 {"speaker": 1,
6 "text": "cheap restaurant",
7 "goals": {"pricerange": "cheap"},
8 "dialog_acts": [{"slots": [["pricerange", "cheap"]], "
    act": "inform"}]}
```

Among them, "speaker = 2" means that this sentence is sent by the dialogue system. "speaker = 1" means that this sentence is input by the user. "goals" is the state value that the DST model should predict, and it is

the ground truth of the DST. "slots" in the "dialogue acts" is the slot value that the slot filling model should predict, and it is the ground truth of the slot filling model.

B.1.3 Output

The output is json format data modified through change_slot.py. In order to be able to run under the DeepPavlov framework, the data still maintains the format of DSTC 2.

```
1 {"speaker": 2,
2 "text": "Hello, welcome to the Cambridge restaurant
    system. You can ask for restaurants by type of
    sentence division terms, type of text summarization
    or type of grammar error correction type. How may I
    help you?",
3 "dialog_acts": [{"act": "welcomemsg", "slots": []}]}
4
5 {"speaker": 1,
6 "text": "abstract restaurant",
7 "goals": {"type of text summarization": "dontcare"},
8 "dialog_acts": [{"slots": [["type of text summarization
    ", "abstract"]], "act": "inform"}]}
```

Using this code, we successfully replaced the important words in the DSTC 2 data set with our own designed slot through complex data structure operations according to the corresponding relationship of the slot.

B.2 Filter high frequency words in DSTC 2

This program of screening high-frequency words is mainly prepared for the operation of replacing high-frequency words later. We hope to replace some of the words representing food and restaurant that have not been replaced in the previous step, so that the model can think that this is a writing aid related data when predicting.

B.2.1 cal_freq.py

B.2.2 Input

The input of this step is the output of the previous step.

Name	cal_freq.py
Type	Python 3.7 script
Version	1
Function	Filter high frequency words in DSTC 2
Argument 1	The file path of modified DSTC 2 dataset
Argument 2	The file path of writing aid dataset
Argument 3	The file path of stop words
Output	the position of high frequency words
Location	http://133.9.48.111:8082/Ni_Zihui/dialogue_systems_writing_aid

Table B.2: Description of cal_freq.py program

B.2.3 Output

We output the position of high-frequency words. The number in this position represents the word order of the sentence in the "text" variable, starting from 0.

```
1 {"speaker": 2, "text": "Hello, welcome to the Cambridge
    restaurant system. You can ask for restaurants by
   type of sentence division terms, type of text
   summarization or type of grammar error correction
   type. How may I help you?", "dialog_acts": [{"act":
   "welcomemsg", "slots": []}]}      4,5,11
2 {"speaker": 1, "text": "abstract restaurant", "goals":
   {"type of text summarization": "dontcare"}, "
   dialog_acts": [{"slots": [["type of text
   summarization", "abstract"]], "act": "inform"}]}   1
```

The use of stop words in the dictionary is to prevent stop words from being screened out when screening. We don't need to let the later language model predict stop words. We can see that "restaurant" has been filtered out in these two sentences. This shows the effectiveness of our code.

B.3 Use language models to replace high frequency words

In this model, we hope to replace the keywords selected in the previous step with the words in writing aid dialogue. The replacement model is a BERT pre-trained language model. We use the BERT language model for finetuning

in the writing aid dialogue, and then understand the position number filtered out earlier as a mask. Enter a whole sentence and output the word mask. We replaced the predicted words with the original words and got a more reasonable dataset.

B.3.1 LM_final.py

Name	LM_final.py
Type	Python 3.7 script
Version	1
Function	Use language models to replace high frequency words
Argument 1	The file with the position of high frequency words
Argument 2	The file path of stop words
Argument 2	The file path of language model weight
Output	the replaced dataset with DSTC 2 format
Location	http://133.9.48.111:8082/Ni_Zihui/dialogue_systems_writing_aid

Table B.3: Description of LM_final.py program

B.3.2 Input

The input of this step is the output of the previous step.

B.3.3 Output

We got a data set that was replaced. The effect of replacement depends entirely on the training effect of the language model. During training, we randomly selected a mask for each sentence, and let the language model train 500 epochs in the writing aid dialogue data set.

```
1 {"speaker": 2, "text": "Hello, welcome to the abstract
    text system. You can ask for text by type of
    sentence division terms, type of text summarization
    or type of grammar error correction type. How may I
    help you?", "dialog_acts": [{"act": "welcomemsg", "
    slots": []}]}
2 {"speaker": 1, "text": "abstract text", "goals": {"type
     of text summarization": "dontcare"}, "dialog_acts":
    [{"slots": [["type of text summarization", "
    abstract"]], "act": "inform"}]}
```

After replacing the language model, we can see that although the sentences are not particularly fluent, most of the words are already writing aid related words. Such sentences can help us train slot filling and DST models to predict the correct value.

B.4 Run the dialogue system

After successfully modifying the data set, we can use the DeepPavlov framework to train the entire dialogue system. We have two tasks: slot filling and DST. We have written the code for how to download, install, replace the data set file, modify the source code and train the model into the jupyter notebook.

B.4.1 demonstration_deeppavlov.ipynb

Name	demonstration_deeppavlov.ipynb
Type	Python 3.7 script, Tensorflow
Version	1
Function	Run the dialogue system
Output	the quantitative result in our modified dataset
Location	http://133.9.48.111:8082/Ni_Zihui/dialogue_systems_writing_aid

Table B.4: Description of demostration_DeepPavlov.ipynb program

B.4.2 Input

The input is our modified data set. This data set still maintains this DSTC 2 format and can run under the framework of DeepPavlov.

B.4.3 Output

We will show the accuracy results of slot filling and DST model. We will show the accuracy results of slot filling and DST. For slot filling tasks, this framework can also calculate the results of precision, recall and F1, but for the convenience of presentation, we only show the results of accuracy.

```
1 2021-07-13 23:04:41.640 WARNING in 'DeepPavlov.core.
    trainers.nn_trainer'['nn_trainer'] at line 341:
    Using NNTrainer for a pipeline without batched
    training
2 2021-07-13 23:04:41.640 INFO in 'DeepPavlov.core.
    trainers.nn_trainer'['nn_trainer'] at line 345: Save
     model to capture early training results
3 {"valid": {"eval_examples_count": 1285, "metrics": {"
    slots_accuracy": 0.979}, "time_spent": "0:00:44"}}
4 {"test": {"eval_examples_count": 1220, "metrics": {"
    slots_accuracy": 0.9811}, "time_spent": "0:00:37"}}
```

This is the result of slot filling. The results obtained on our modified data set are similar to the results obtained on the original DSTC 2 data set, which can prove the validity of our data.

```
1 2021-07-13 23:14:44.309 INFO in 'DeepPavlov.models.
    go_bot.policy.policy_network'['policy_network'] at
    line 86: INSIDE PolicyNetwork init(). Initializing
    PolicyNetwork from checkpoint.
2 2021-07-13 23:14:44.310 INFO in 'DeepPavlov.core.models
    .tf_model'['tf_model'] at line 51: [loading model
    from ~/.DeepPavlov/models/gobot_DSTC 2/model/policy]
3 {"valid": {"eval_examples_count": 575, "metrics": {"
    per_item_action_accuracy": 0.5001}, "time_spent":
    "0:00:15"}}
4 {"test": {"eval_examples_count": 576, "metrics": {"
    per_item_action_accuracy": 0.5076}, "time_spent":
    "0:00:15"}}
```

This is the result of DST. The result obtained on our modified data set is slightly worse than the effect on the original DSTC 2. The reason may be that we have more sentences that are not fluent, which affects the training and prediction of the model.

B.5 Example of an actual dialogue

This is a real example of using a dialogue system to generate dialogue. In this example, we used five sentences to test the dialogue system. After testing, we believe that our dialogue system is effective.

```
1 x::hi
```

```
2 >> ['Hello, welcome to the writing aid system. You can
     do many kinds of modification of the paragraph here.
      Do you want a grammar check first?']
3 x::I want to recognize the number of total words.
4 >> ["Okay, I'll do it for you."]
5 x::I hope you could tell me the average number of words
     .
6 2021-07-15 16:21:53.251 INFO in 'DeepPavlov.models.
     go_bot.tracker.dialogue_state_tracker'['
     dialogue_state_tracker'] at line 166: Made api_call
     with {}, got 109 results.
7 >> ["okay, i'll do it for you.", "Okay, I'll do it for
     you."]
8 x::Please demonstrate it with the result of text
     summarization with opening sentence.
9 >> ["Okay, I'll do it for you."]
10 x::Could you give me a cup of coffee?
11 >> ['Sorry, I can't do it for you.']
12 x::I want you to check the plagiarism.
13 2021-07-15 16:27:36.489 INFO in 'DeepPavlov.models.
     go_bot.tracker.dialogue_state_tracker'['
     dialogue_state_tracker'] at line 166: Made api_call
     with {}, got 109 results.
14 >> ['sorry, i can't do it for you.', 'Sorry, I can't do
      it for you.']
```

References

Bird, S., Dale, R., Dorr, B., Gibson, B., Joseph, M., Kan, M.-Y., Lee, D., Powley, B., Radev, D., and Tan, Y. F. (2008). The ACL Anthology reference corpus: A reference dataset for bibliographic research in computational linguistics. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Bohus, D. and Rudnicky, A. I. (2009). The ravenclaw dialog management framework: Architecture and systems. *Comput. Speech Lang.*, 23:332–361.

Budzianowski, P., Wen, T.-H., Tseng, B.-H., Casanueva, I., Ultes, S., Ramadan, O., and Gašić, M. (2018a). 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.

Budzianowski, P., Wen, T.-H., Tseng, B.-H., Casanueva, I., Ultes, S., Ramadan, O., and Gašić, M. (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.

Burtsev, M., Seliverstov, A., Airapetyan, R., Arkhipov, M., Baymurzina, D., Bushkov, N., Gureenkova, O., Khakhulin, T., Kuratov, Y., Kuznetsov, D., Litinsky, A., Logacheva, V., Lymar, A., Malykh, V., Petrov, M., Polulyakh, V., Pugachev, L., Sorokin, A., Vikhreva, M., and Zaynutdinov, M. (2018a). DeepPavlov: Open-source library for dialogue systems. In *Proceedings of ACL 2018, System Demonstrations*, pages 122–127, Melbourne, Australia. Association for Computational Linguistics.

Burtsev, M., Seliverstov, A., Airapetyan, R., Arkhipov, M., Baymurzina, D.,

Bushkov, N., Gureenkova, O., Khakhulin, T., Kuratov, Y., Kuznetsov, D., Litinsky, A., Logacheva, V., Lymar, A., Malykh, V., Petrov, M., Polulyakh, V., Pugachev, L., Sorokin, A., Vikhreva, M., and Zaynutdinov, M. (2018b). DeepPavlov: Open-source library for dialogue systems. In *Proceedings of ACL 2018, System Demonstrations*, pages 122–127, Melbourne, Australia. Association for Computational Linguistics.

Chang, Y.-C., Chang, J. S., Chen, H.-J., and Liou, H.-C. (2008). An automatic collocation writing assistant for taiwanese efl learners: A case of corpus-based nlp technology. *Computer Assisted Language Learning*, 21(3):283–299.

Cieliebak, M., Deriu, J. M., Egger, D., and Uzdilli, F. (2017). A Twitter corpus and benchmark resources for German sentiment analysis. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*, pages 45–51, Valencia, Spain. Association for Computational Linguistics.

Danescu-Niculescu-Mizil, C. and Lee, L. (2011). Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics, ACL 2011*.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (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.

Dinan, E., Roller, S., Shuster, K., Fan, A., Auli, M., and Weston, J. (2019). Wizard of wikipedia: Knowledge-powered conversational agents. In *International Conference on Learning Representations*.

Eckert, W., Levin, E., and Pieraccini, R. (1997). User modeling for spoken dialogue system evaluation. In *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, pages 80–87.

Fainberg, J., Krause, B., Dobre, M., Damonte, M., Kahembwe, E., Duma, D., Webber, B. L., and Fancellu, F. (2018). Talking to myself: self-dialogues as data for conversational agents. *CoRR*, abs/1809.06641.

Feng, S., Wan, H., Gunasekara, C., Patel, S., Joshi, S., and Lastras, L. (2020). doc2dial: A goal-oriented document-grounded dialogue dataset.

In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8118–8128, Online. Association for Computational Linguistics.

Heinroth, T., Denich, D., and Schmitt, A. (2010). Owlspeak - adaptive spoken dialogue within intelligent environments. volume 43, pages 666 – 671.

Henderson, M., Thomson, B., and Williams, J. D. (2014a). 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.

Henderson, M., Thomson, B., and Williams, J. D. (2014b). 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.

Huebner, T. (2015). Academic writing for graduate students: Essential tasks and skills, john m. swales and christine b. feak. 2012. ann arbor, mi: University of michigan press (418 pages). *MANUSYA*, 18:113–115.

Jung, S., Lee, C., and Lee, G. (2008). Dialogstudio: A workbench for data-driven spoken dialog system development and management. *Speech Communication*, 50:697–715.

Li, J., Jia, R., He, H., and Liang, P. (2018). Delete, retrieve, generate: a simple approach to sentiment and style transfer. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1865–1874, New Orleans, Louisiana. Association for Computational Linguistics.

Li, K.-C., Chang, M., and Wu, K.-H. (2020). Developing a task-based dialogue system for english language learning. *Education Sciences*, 10(11).

Li, Y., Su, H., Shen, X., Li, W., Cao, Z., and Niu, S. (2017). DailyDialog: A manually labelled multi-turn dialogue dataset. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 986–995, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Lison, P. and Kennington, C. (2016a). OpenDial: A toolkit for developing spoken dialogue systems with probabilistic rules. In *Proceedings of ACL-2016 System Demonstrations*, pages 67–72, Berlin, Germany. Association for Computational Linguistics.

Lison, P. and Kennington, C. (2016b). OpenDial: A toolkit for developing spoken dialogue systems with probabilistic rules. In *Proceedings of ACL-2016 System Demonstrations*, pages 67–72, Berlin, Germany. Association for Computational Linguistics.

Lison, P. and Tiedemann, J. (2016). OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 923–929, Portorož, Slovenia. European Language Resources Association (ELRA).

Liu, B., Tür, G., Hakkani-Tür, D., Shah, P., and Heck, L. P. (2017). End-to-end optimization of task-oriented dialogue model with deep reinforcement learning. *CoRR*, abs/1711.10712.

Malchanau, A., Petukhova, V., and Bunt, H. (2018). Towards continuous dialogue corpus creation: writing to corpus and generating from it. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Miller, A., Feng, W., Batra, D., Bordes, A., Fisch, A., Lu, J., Parikh, D., and Weston, J. (2017a). 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.

Miller, A., Feng, W., Batra, D., Bordes, A., Fisch, A., Lu, J., Parikh, D., and Weston, J. (2017b). 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.

Nguyen, T. and Shcherbakov, M. (2021). Enhancing rasa nlu model for vietnamese chatbot. *International Journal of Open Information Technologies*, 9(1):31–36.

Stasaski, K., Kao, K., and Hearst, M. A. (2020). CIMA: A large open access dialogue dataset for tutoring. In *Proceedings of the Fifteenth Workshop on*

Innovative Use of NLP for Building Educational Applications, pages 52–64, Seattle, WA, USA Online. Association for Computational Linguistics.

Ultes, S., Rojas-Barahona, L. M., Su, P.-H., Vandyke, D., Kim, D., Casanueva, I., Budzianowski, P., Mrkšić, N., Wen, T.-H., Gašić, M., and Young, S. (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.

Wei, W., Le, Q., Dai, A., and Li, J. (2018a). AirDialogue: An environment for goal-oriented dialogue research. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3844–3854, Brussels, Belgium. Association for Computational Linguistics.

Wei, W., Le, Q., Dai, A., and Li, J. (2018b). AirDialogue: An environment for goal-oriented dialogue research. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3844–3854, Brussels, Belgium. Association for Computational Linguistics.

Yan, Z., Duan, N., Chen, P., Zhou, M., Zhou, J., and Li, Z. (2017). Building task-oriented dialogue systems for online shopping. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1).

Zhang, S., Dinan, E., Urbanek, J., Szlam, A., Kiela, D., and Weston, J. (2018). Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2204–2213, Melbourne, Australia. Association for Computational Linguistics.

Zhu, Q., Zhang, Z., Fang, Y., Li, X., Takanobu, R., Li, J., Peng, B., Gao, J., Zhu, X., and Huang, M. (2020). Convlab-2: An open-source toolkit for building, evaluating, and diagnosing dialogue systems.