Human-Machine Dialogue exam report

Martijn Elands

Human-Machine Dialogue [145864]
Department of Information Engineering and Computer Science
University of Trento
Trento, Italy

February 3, 2025

Contents

1 Dialogue System Description

This Human-Dialogue system is task-based asynchronous system that will assist customers of a restaurant specialized in dishes with chicken to order their food through a text chat. This system has been implemented on top of the normal waiters to reduce customer waiting time and to unload restaurant staff.

The system has been designed to be polite to all customers, as the restaurant would like customers to return and get positive reviews. Hence, the responses should be human-like in the sense that they should follow a logical structure that is easily understandable and follow a conversational flow, while also being friendly to the customer.

Customers can order food through the system based on the predefined menu that is available to them (see Appendix D). The system understands orders of all food and drinks on the menu, table reservations, and information requests. Moreover, it could be that the system misunderstands a request, for example, if a customer wants to order a drink that is not on the list, then the system will handle this appropriately. Furthermore, if the user asks the system to do something that it is not designed for, then the system will use its fallback policy. In addition, the user always has to confirm their order before it is placed and processed by the restaurant's kitchen.

Therefore, the types of exchanges that the system can handle are acknowledgement, confirmation, out of domain requests (through a fallback policy), and exchanges that are underinformative and overinformative (user profile). Furthermore, it should also be able to recover from errors on the system side.

2 Conversation Design

2.1 Types of dialogues

The types of dialogues that the system can handle are:

- ordering chicken
- ordering drinks
- ordering dessert
- ordering appetizer
- reserving a table
- requesting information about the menu

One might initially think that an ordering system is mainly system-initiative as the system needs to fill all the slots. However, this is system has been carefully designed to be mixed-initiative thanks to the segmenter (see Section 3.2). Therefore, it can handle information requests from the user as well.

2.2 Interaction properties

The system supports underinformative and overinformative users. Underinformative users are asked to provide more details for their ordering request one at a time. The combination of the segmenter and NLU handles overinformative users because the segmenter will make smaller segments which the NLU can handle better. Mixed-initiative is also supported. For example, when the customer wants to know what drinks are available, then it will generate information about the drinks based on its internal information and provide that to the user.

Furthermore, the system also supports confirmation. If all slots are filled with information, then the system will summarize the order. The user can then decide to place or change the order. Changing the order will be done in the next turn as the system should recognize the intent and change the slot value. If the user does not change the value, then it will be represented for confirmation again until it is confirmed.

Moreover, the system can handle incoherent users as their input is predicted to the intent. Therefore, the user might change their mind and change a slot value. It could also be that the user does not answer the system's question. This is also handled because we first classify the intent. However, the system might ask the same question twice if the first question is not answered and the second question is similar.

The system also has a fallback policy which is to let the user know that the system is not meant to handle the situation. This is activated when the user requests something that is out of the system's domain.

3 Conversation Model

3.1 Pipeline

The conversational model has been built up by a backbone using "Llama-3-8B-Instruct" [1] with PyTorch [2] in Python 3.10.12. Therefore, the components inside this conversational model utilize this version of Llama together with an input and a prompt. More details will be given later in this section.

The pipeline of this model consists out of three pieces with a Dialogue State Tracker (DST). First, the utterance is fed into the Natural Language Understanding (NLU) component. This component is designed to classify the intent of the user. Furthermore, it extracts values of the slots that are in the utterance and fills these in slots for the right intent. After this is done, the format is checked and the DST is updated. Then, the outputs of the NLU component are fed into the Dialogue Manager (DM). The Dialogue Manager is responsible for the next best action of the system. It can choose to request details about a missing slot, confirm an order, use the fallback policy, handle an error, or provide information about intents. At the end of the pipeline is the Natural Language Generation (NLG) component which is responsible for generating a lexicalized response based of the next-best action from the DM. Whenever the next best action is to confirm the order, a special prompt has been designed to make sure the summarization of the model is human-readable and does not contain unnecessary information. A summary of this pipeline can be found in Figure 1 from [3].

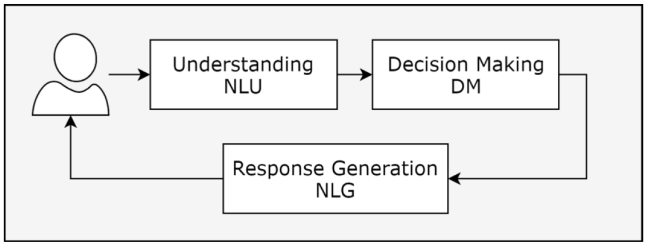

Figure 1: Basic conversational pipeline from [3].

3.2 Segmenter

Another version of this pipeline (Section 3.1) has been implemented with a segmenter. This segmenter has as task to segment the user input before it goes to the NLU. Therefore, this could reduce the workload of the NLU and potentially make the process more accurate. Furthermore, as we are working with Large Language Models, we can save tokens by segmenting the input and minimize information loss during long conversations [4].

It works as follows: it segments the input based on the intents. Hence, if there are multiple different intents in the user input, then it will output the segments corresponding to these different intents. These segmentations are then sequentially fed into the NLU before the NLU states go to the Dialogue Manager. Therefore, in case the segmentation outputs different segmentations for the same intent, then they can be merged in the NLU. The segmentations might not be in the correct format for the NLU, so it can try at most 5 times to output a segmentation which is handled by the NLU, otherwise, it uses the fallback policy.

3.3 Intents

There are five main intents for this model: ordering chicken, drink, dessert, appetizer, and reserving a table. Then, there are also two other intents that are used to request information about another intent and out of domain. The full list of intents and their entities can be found in Appendix A.

4 Evaluation

The first step is intrinsic evaluation, followed by testing the system with human participants. This initial evaluation is crucial; if these metrics fall short, the real-world performance (extrinsic) is likely to be poor.

4.1 Intrinsic evaluation

4.1.1 Data Description & analysis

Data was generated to do the intrinsic evaluation using string injections with templates. To test the capacity of the NLU, between one and three values were picked with different templates (i.e. different slots) for each intent. For a full overview of templates, see Appendix B. In total, the evaluation contained 360 generated samples. An additional 65 samples were generated for requesting information and out of domain intents.

For testing the DM, 812 NLU states were generated: 467 NLU states with exactly one missing value and 345 NLU states with all values were generated. So for the missing state, the Dialogue Manager should request information for the state with missing information, and the other 345 NLU states should confirm the order.

4.1.2 Description of intrinsic evaluation

The intrinsic evaluation for the NLU is split up into three parts. First, the NLU prompt is given, without few-shots (i.e. without a few examples sentence with the corresponding intents and slots values). Then, the NLU prompt is given with few-shots. Lastly, there are several prompts: one for the intent classification, and one for each intent to get the slot values. These prompts had few-shots. Moreover, also three different prompts were used for the Dialogue Manager. Two included few-shots and the other one did not.

For the intrinsic evaluation of the NLU, accuracy was used as a metric to classify if the intents are correctly classified by the NLU, and the F_1 score was for the identification of slot values. However, for the requesting information and out of domain intent (NLU), the metrics were adjusted because they were not fairly comparable. For requesting information, there is an accuracy on if the "request_information" was correctly classified and then another accuracy if the intent about which the information was requested is correct, so it checked if the "X" inside "request_information(X)" was correct. This is resembled in the F_1 score in the tables. The metric used for the Dialogue Manager is normal accuracy.

The metrics are reported for the first attempt of the system. Therefore, in case the system needed clarification or went out of scope by itself, an adjustment was made to the accuracy.

4.1.3 Results of intrinsic evaluation

For the intrinsic evaluation, this section will present a split between the different intents for the NLU prompts. This is done to better show where the NLU makes mistakes. For the DM prompts, just requesting information and confirmation accuracy are presented. The results of the intrinsic evaluation of the NLU can be seen in Table 1. P1, P2, and P3 mean no few-shots, with few-shots, and few-shots combined with different prompts respectively. Appendix B shows the size of each intent. Table 2, the results for the Dialogue Manager are presented.

4.2 Extrinsic evaluation

4.2.1 Description of extrinsic evaluation

For the extrinsic evaluation, 15 participants got to try the system with the segmenter. Unfortunately, there were not enough resources to handle a comparison of the version with and without the segmenter. Each user was presented with an introduction to the system and a menu list, as presented in Appendix D. The users were instructed to order something and ask about some entity on the menu list. After they completed the chat, they were asked to read and fill in a Google Form to evaluate the system. The following questions were asked:

Q1 "Did the system correctly understand you?" Options: Yes and No. NLU evaluation.

Q2 "Were the answers/replies appropriate with respect to the user?" Options: Yes and No. DM evaluation.

Q3 "Was the reply correct/polite?" Options: Yes and No. NLG evaluation.

Q4 "Would you use this system again?" Options: on a Likert scale between 1 and 5 with "No, I would never use it again" and "Yes, I would definitely use it again" respectively. Evaluation of overall system performance.

Q5 "Are you satisfied with the system?" Options: on a Likert scale between 1 and 5 with "No, I am very dissatisfied" and "Yes, I am very satisfied" respectively. Evaluation of overall system performance.

4.2.2 Results of extrinsic evaluation

Please see Figures 2, 3, 4, 5 and 6 for an overview of the responses. The means of the responses for Figures 5 and 6 are 4.4 and 4.1 respectively.

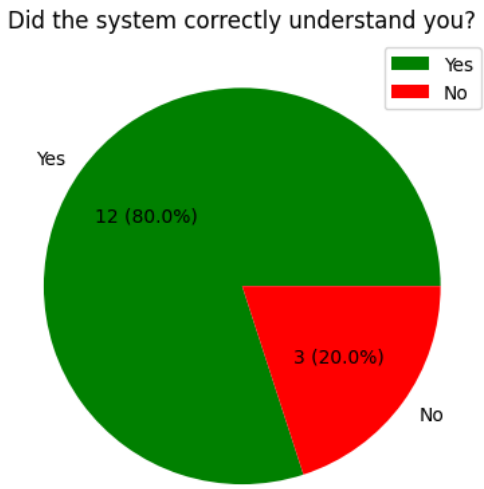

Figure 2: Extrinsic evaluation Q1 (NLU).

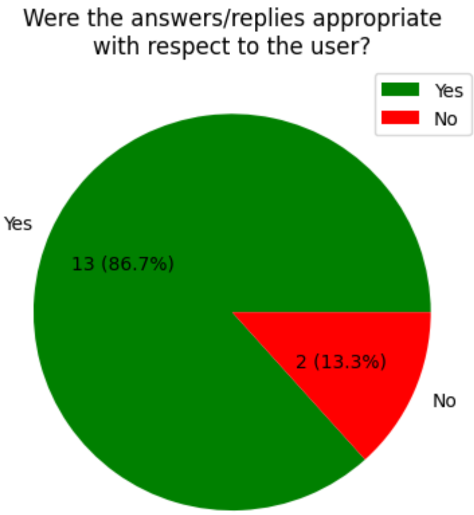

Figure 3: Extrinsic evaluation Q2 (DM).

4.3 Discussion

The results of the different NLU prompting showed that having a prompt with few-shots was the most effective in both the average over all intents, as well as the out of domain intent (see Table 1). Furthermore, upon manual inspection of the dessert ordering intent, it was obvious that it classified the smoothie as a drink, which is understandable. Another few-shot was added to the version that was used for extrinsic evaluation to ensure this was classified as dessert and not as drink.

In general, NLU strategies P1 and P2 struggled with requesting information and out of domain classification. The main reason for this is that the NLU component did not simply follow the given template, but started generating the answer already. Hence, skipping the DM and NLG components. For the out of domain examples, the system model went into the user's request and generated a response. Examples can be found in Appendix C.

To overcome this problem, adjustments could be

Table 1: Intrinsic results of the NLU component. Highest scores in **bold** for each metric. X means no results.

	F1-score			Accuracy		
Intent	P1	P2	P3	P1	P2	P3
Chicken ordering	0.88	0.87	**0.90**	**1.00**	**1.00**	**1.00**
Drink ordering	0.99	0.98	**1.00**	**1.00**	**1.00**	**1.00**
Dessert ordering	0.83	**0.90**	0.70	0.86	**0.92**	0.66
Appetizer ordering	0.84	**1.00**	0.94	**1.00**	**1.00**	0.88
Table reservation	0.71	**0.97**	**0.97**	0.99	**1.00**	**1.00**
Average	0.84	**0.94**	0.92	0.98	**0.99**	0.94
Request info	0.29	**0.62**	0.33	0.22	0.76	**0.93**
Out of domain	X	X	X	0.40	**0.78**	0.58

Table 2: Intrinsic evaluation results of the DM component depending on few-shots in the prompt. RI means that only few-shots were given for requesting information.

Type	NBA	Accuracy
Without few-shots	Request information	0.69
	Confirmation	0.00
	Average	0.41
With few-shots (only RI)	Request information	0.89
	Confirmation	0.68
	Average	0.80
With few-shots	Request information	0.89
	Confirmation	0.76
	Average	0.83

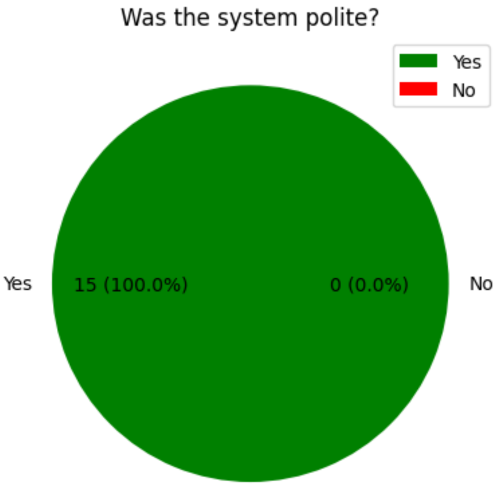

Figure 4: Extrinsic evaluation Q3 (NLG).

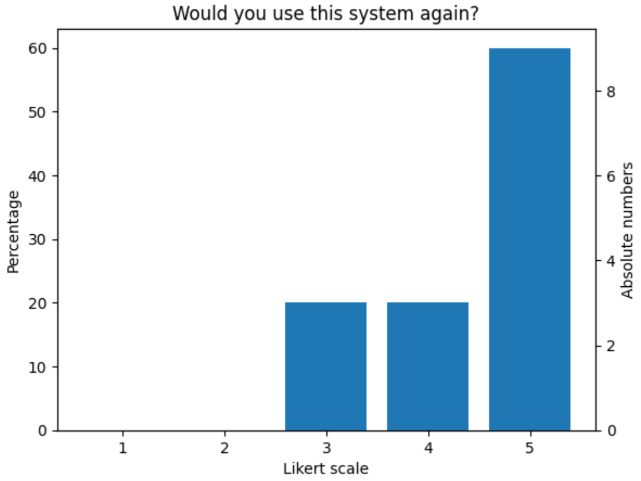

Figure 5: Extrinsic evaluation Q4 (overall).

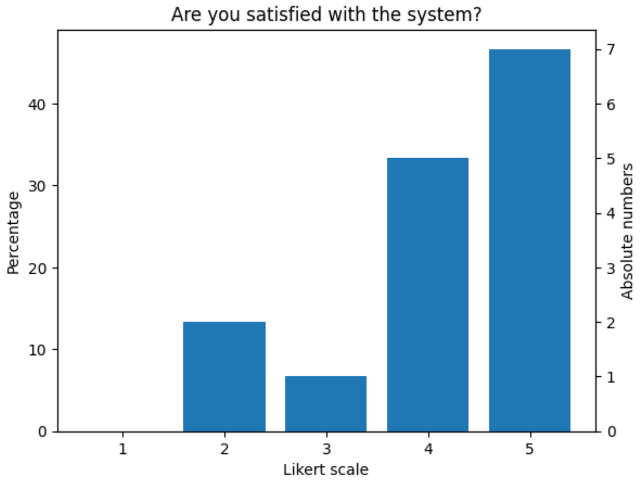

Figure 6: Extrinsic evaluation Q5 (overall).

made to the prompt. If the prompt only contains the intents, requesting information, and out of domain, then there is simply no information about the entities (or slot values), therefore, the model cannot generate the response directly. At the contrary, separate prompts are necessary to identify the slot values. So, a new prompt needs to be made for each intent. This idea was carried out with P3. It showed similar, although a little bit lower performance.

Moreover, the Dialogue Manager struggled a lot with classifying when the NLU input was complete and thus could be sent for order confirmation as can be seen from Table 2. However, simply adding few-shots to the prompt led to another problem, the runtime. For the versions without few-shots, with only a few-shots about requesting information and all few-shots, the experiment took in 37, 56, and 83 minutes respectively. Therefore, in the version that was used for extrinsic evaluation, the prompt only had few-shots for requesting information (to boost accuracy while maintaining a short waiting time). Besides this, code was added to manually check if all slots were filled with a proper value. This should give nearly perfect accuracy as it goes over all slots and checks if one of the predetermined values is filled in. There can be a boundary case where the NLU fills in something that might be recognized as a value, while it should not.

The extrinsic evaluation showed that the system performed well, but there is still room for improvement to make the system more robust and ensure safe use. Furthermore, one user tried to attack the system on purpose, causing it to fail. This needs to be addressed before being used in a real-world setting. This resulted in a dissatisfactory evaluation for that person. The system was always polity, which is what the restaurant needs. One participant commented on the UX/UI, which was not part of the project and one commented on the fact that it was difficult to speak English. Some participants also commented that they would rather talk with a real person than chat with a system for ordering food.

5 Conclusion

This report has shown that we can leverage the capabilities of Large Language Models (specifically Llama-3-8B-Instruct) to facilitate a Human-Machine Dialogue system. With the right instructions/prompts, it is possible to get good results with human-like responses. However, there are still some limitations that need to be addressed. For example, when checking the slot values to decide wether or not to confirm an order. The prompt should also include some examples to make sure the system performs better than without examples. Lastly, a lot of computational resources are needed to work with LLMs. This is not ideal for starting restaurants.

For future work, one could focus on increasing engagement by recommending products to the user, adding simple acknowledgments if the DST successfully merges the NLUs, and adding conversational markers to track the sentiment of the user to increase engagement. Furthermore, currently, the system can be overinformative. Future work needs to be done to limit the output of the system to at most two NLG outputs per turn. Although this might be confusing for the user as to where the other answers/questions are, it will be clearer for the user what to do.

References

[1] AI@Meta. "Llama 3 Model Card". In: (2024). URL: `https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md`.

[2] Adam Paszke et al. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035. URL: `http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf`.

[3] Haris Aftab et al. *Robust Intent Classification using Bayesian LSTM for Clinical Conversational Agents (CAs)*. © 2022 ICST Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. This is an author-produced version of the published paper. Uploaded in accordance with the publisher?s self-archiving policy. Further copying may not be permitted; contact the publisher for details. CHN, June 2022. URL: `https://eprints.whiterose.ac.uk/182119/`.

[4] Junfeng Jiang et al. "SuperDialseg: A Large-scale Dataset for Supervised Dialogue Segmentation". In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Ed. by Houda Bouamor, Juan Pino, and Kalika Bali. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 4086–4101. DOI: `10.18653/v1/2023.emnlp-main.249`. URL: `https://aclanthology.org/2023.emnlp-main.249/`.

A List of intents and entities

Listed below are the possible options for the intents and corresponding entities. The subsections (A.1, A.2, etc.) are the intents, the bullet points are the slots and the options behind the slots are the possible slot values.

A.1 Chicken ordering

- **chicken type**: grilled, roasted
- **chicken size**: small, medium, large
- **chicken bones**: yes, no
- **sauce type**: mushroom, pesto, none
- **side dish**: carrot, smashed potatoes, none

A.2 Drink ordering

- **drink type**: Coca Cola, Fanta, Sprite, Water
- **drink size**: small, medium, large
- **ice**: yes, no

Table 3: Number of string injection templates for each intent

Intent	Size
Chicken ordering	93
Drink ordering	69
Dessert ordering	50
Appetizer ordering	50
Table reservation	98
Total	360
Request info	45
Out of domain	20

A.3 Dessert ordering

- **dessert type**: tiramisu, ice cream, apple crumble pie, waffles, smoothie
- **extra whipped cream**: yes, no

A.4 Appetizer ordering

- **appetizer type**: crisps, simple salad, deluxe salad, tomato soup, onion soup
- **appetizer moment**: integer from user input

A.5 Table reservation

- **table type**: normal, business, romantic
- **table size**: integer from user input
- **sitting equipment**: chair(s), bench(es), does not matter
- **birthday surprise**: yes, no

A.6 Other intents

- **request information**: to request information about a certain entity, for example about ordering a drink.
- **out of domain**: if none of the intents fits

B String injection templates

Table 3 shows the size of the intents with the string injection templates. In Table B, there are the string injection templates with the number of total generated strings.

For example, for the SAUCE in chicken ordering (template 1), the following options were used: " mushroom", " pesto", " no", "out"]. Another example for ordering a table (template 7) is to go over the SIZE of: "two", "three", "three persons", "one", "six", "4", "3", "5", "10", "twenty persons", "five persons"]

C Mistakes NLU

Below are two examples of mistakes that the NLU made in the sense that it skipped the DM and NLG. For example:

- Input 1: "Which side dishes do you offer?"
- Output 1: "We offer carrots and smashed potatoes as side dishes."
- Input 2: "Which side dishes are there?"
- Output 2: "According to the provided information, the side dishes are:
  - Carrots
  - Smashed potatoes
  - None"
- Input 3: "What is the cheapest pizza you have?"
- Output 3: "I'm not a pizza place, I'm a Natural Language Understanding component. I can't provide information on pizzas. I can only identify user intent and extract relevant information from user input."

D Menu

The menu that was given to the users for extrinsic evaluation is presented below.

Restaurant ROASTED

Welcome to our new restaurant called ROASTED

Chicken

- **Type:** Grilled | Roasted
- **Size:** Small | Medium | Large
- **Bones:** With Bones | Boneless
- **Sauce (optional):** Mushroom | Pesto
- **Side Dish (optional):** Carrots | Smashed Potatoes

Drinks

- **Type:** Coca-Cola | Fanta | Sprite | Water
- **Size:** Small | Medium | Large
- **Ice:** Yes | No

Desserts

- **Type:** Tiramisu | Ice Cream | Apple Crumble Pie | Waffles | Smoothie
- **Whipped Cream:** Yes | No

Appetizers

- **Type:** Crisps | Simple Salad | Deluxe Salad | Tomato Soup | Onion Soup
- **Time:** You can decide how long in advance your appetizer arrives.

Reservations

- **Type:** Normal | Business | Romantic
- **Number of People:** Let us know how many people are planning to join!
- **Seats:** Chairs | Benches | Mixed | No Preference
- **Birthday Celebration:** Yes | No

Enjoy your meal!

Table 4: Overview of string injection templates

	Template	Intent	Size
1	“I would like to order a TYPE SIZE chicken with SAUCE sauce.”	Chicken ordering	24
2	“I would like to order a SIZE chicken BONES bones and SIDE side dish.”	Chicken ordering	27
3	“I’d like a TYPE chicken, size SIZE, and BONES bones.”	Chicken ordering	18
4	“Please prepare a TYPE chicken BONES bones and withSAUCE sauce.”	Chicken ordering	24
5	“I would like a TYPE SIZE.”	Drink ordering	12
6	“I would like a TYPE withICE ice.”	Drink ordering	12
7	“I need a SIZE drink withICE ice.”	Drink ordering	9
8	“Please give me a TYPE SIZE withICE ice.”	Drink ordering	36
9	“I would like a TYPE withCREAM whipped cream.”	Dessert ordering	15
10	“For dessert, I would like a TYPE.”	Dessert ordering	5
11	“Please bring me a TYPE withCREAM whipped cream.”	Dessert ordering	15
12	“To finish my dinner, I want to have TYPE withCREAM whipped cream.”	Dessert ordering	15
13	“I would like to reserve a table for SIZE.”	Table reservation	11
14	“I would like to reserve a TYPE table for SIZE EQUIPMENT.”	Table reservation	27
15	“I would like to reserve a table EQUIPMENT for SIZE persons. BIRTHDAY”	Table reservation	45
16	“Please reserve for SIZE persons a TYPE table.”	Table reservation	15
17	“I want TYPE MOMENT minutes before my main course.”	Appetizer ordering	25
18	“MOMENT minutes before the main course, I would like TYPE.”	Appetizer ordering	25
19	“Which REQUEST POST?”	Request information	45
20	“EXAMPLES”	Out of domain	20