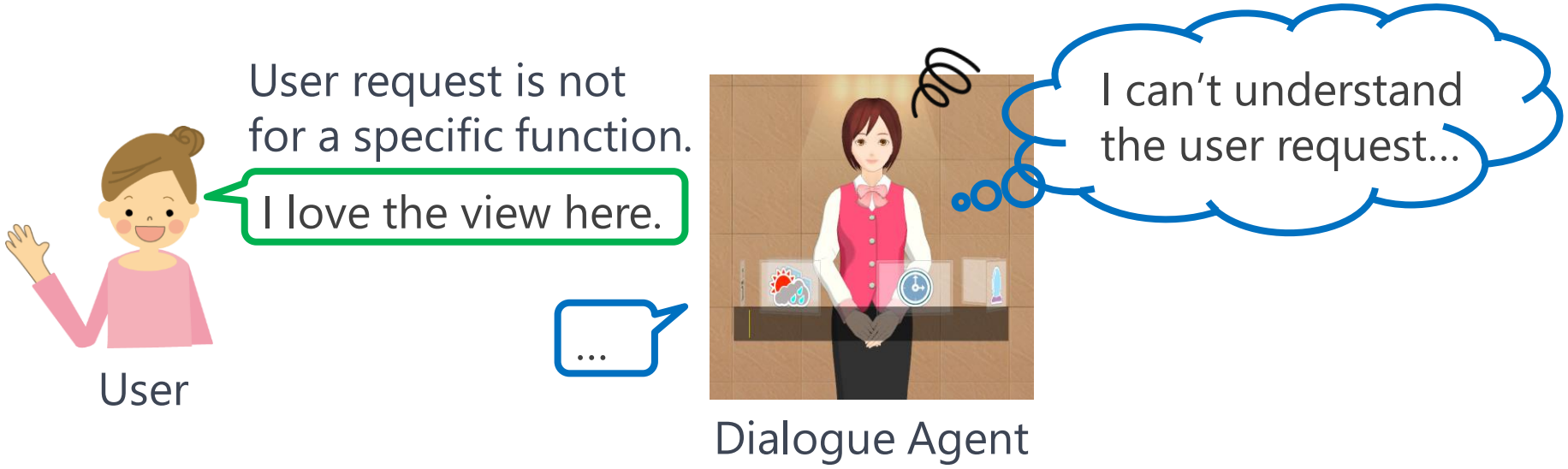
ARTA: Collection and Classification of Ambiguous Requests and Thoughtful Actions

Shohei Tanaka[1,2,3], Koichiro Yoshino[3,1,2], Katsuhito Sudoh[1,2], Satoshi Nakamura[1,2]

[1]Nara Institute of Science and Technology (NAIST)
[2]Center for Advanced Intelligence Project (AIP), RIKEN
[3]Guardian Robot Project (GRP), R-IH, RIKEN

SIGDIAL 2021

Existing Task-Oriented Dialogue Systems

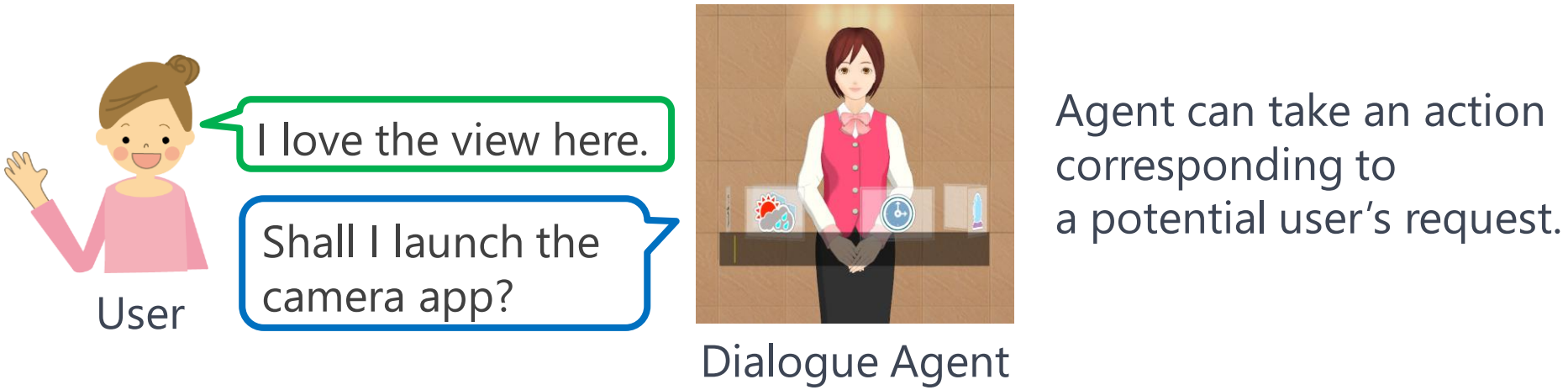


Respond to user requests using pre-defined APIs.

Assume that the user requests are clear and explicit.

Unable to generate appropriate actions when the requests are ambiguous.

Research Objective: Thoughtful Dialogue Agent



Agent can take an action corresponding to a potential user's request.

Thoughtful human concierge can take a "thoughtful action" as shown in the figure.

Aim to build a thoughtful dialogue agent that enables to take thoughtful actions to ambiguous requests.

Corpus Collection Method

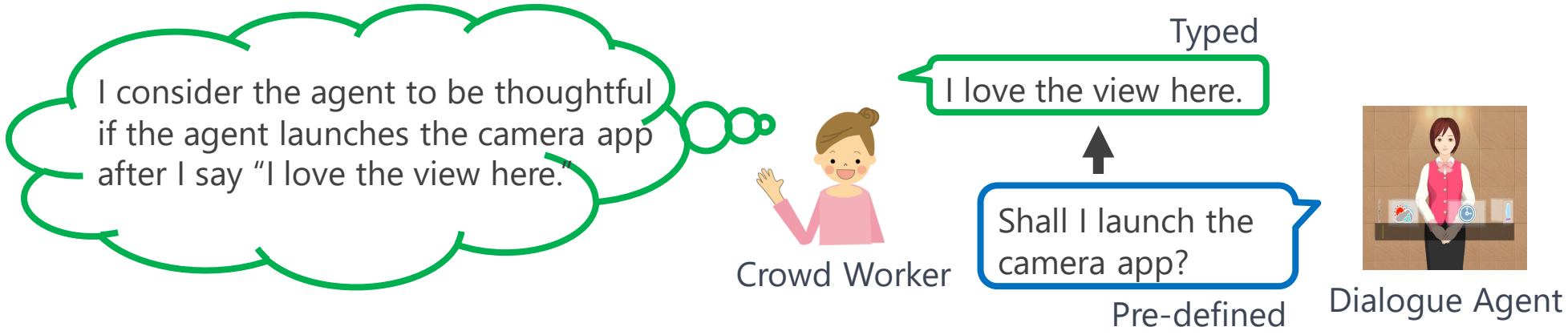
Problems of general WoZ (Wizard of Oz) method:

- Taking thoughtful actions is difficult even for humans.
- System actions are limited by functions such as API.

↓

Easier to ensure the quality of the corpus when system actions are defined.

Crowd workers type preceding requests that defined actions can be considered thoughtful.



Situation settings

Domain is sightseeing in Kyoto.

Interaction between a user and a dialogue agent on a smartphone.

All dialogues consist of **one request and one action**.

System actions are defined in advance based on APIs.

Function	Category	# of category
spot search	park, shrine, etc.	30
restaurant search	sushi, shaved ice, etc.	30
app launch	camera, map, etc.	10

Action form examples:

- Spot search: “Shall I search for a park around here?”
- Restaurant search: “Shall I search for a sushi around here?”
- App launch: “Shall I launch the camera application?”

Collection Results

Function	Ave. length	# requests
spot search	13.44 (+-4.69)	11,670
restaurant search	14.08 (+-4.82)	11,670
app launch	13.08 (+-4.65)	3,890
all	13.66 (+-4.76)	**27,230**

Corpus statistics

User request (collected)	System action (pre-defined)
I'm sweaty and uncomfortable.	Shall I search for a hot spring around here?
I'm bored of Japanese food.	Shall I search for meat dishes around here?
Nice view.	Shall I launch the camera application?

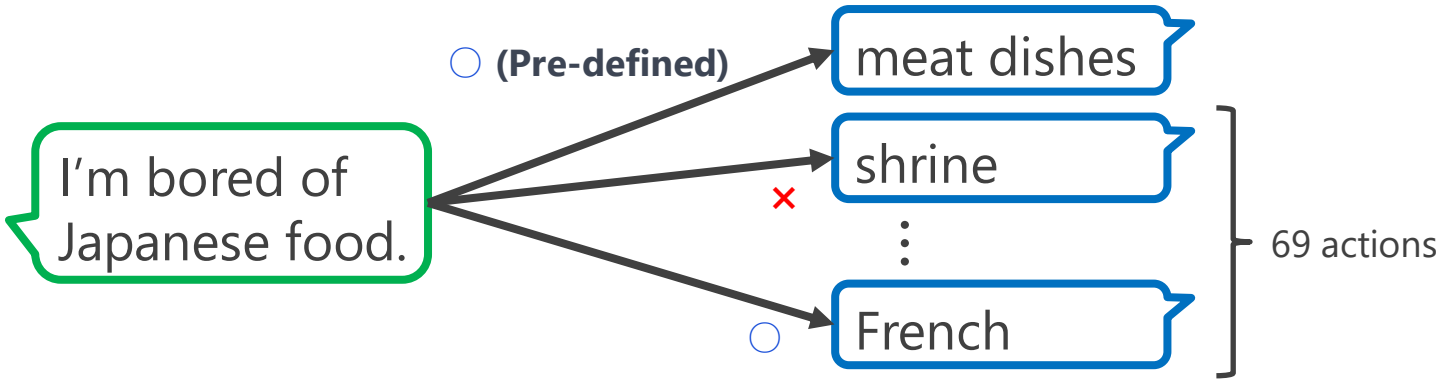
Examples of user requests

Split the data for each category in the ratio of
train data : valid data : test data = 8:1:1 = 21,784:2,730:2,730.

Multi-Class Classification Problem

Because the collected user requests are ambiguous, some of the 69 unannotated actions can be thoughtful.

e.g. For a request "**I'm bored of Japanese food**," suggesting **any type of restaurant other than Japanese** can be thoughtful.



Labeling all combinations of user requests and system actions is costly and impractical.

⬇

Completely annotated all combinations in the **test data** with crowdsourcing.

Resultant Corpus

Data	User request (collected)	System action (pre-defined)	System action (additionally annotated)
train	I'm sweaty and uncomfortable.	hot spring (Shall I search for a hot spring around here?)	- (**no annotation**)
test	I'm bored of Japanese food.	meat dishes (Shall I search for meat dishes around here?)	Chinese, French, etc.

Examples of dialogue with additional annotation

No train data has additional annotation.

Only test data has additional annotation.

Additional actions could also be regarded as thoughtful.

Additional Annotation Statistics

Function	# added categories
spot search	8.45 (+-7.34)
restaurant search	**9.81** (+-7.77)
app launch	5.06 (+-8.48)
all	8.55 (+-**7.84**)

added action categories

Heat map of combination

Restaurant has the highest average because it associates with diverse categories.

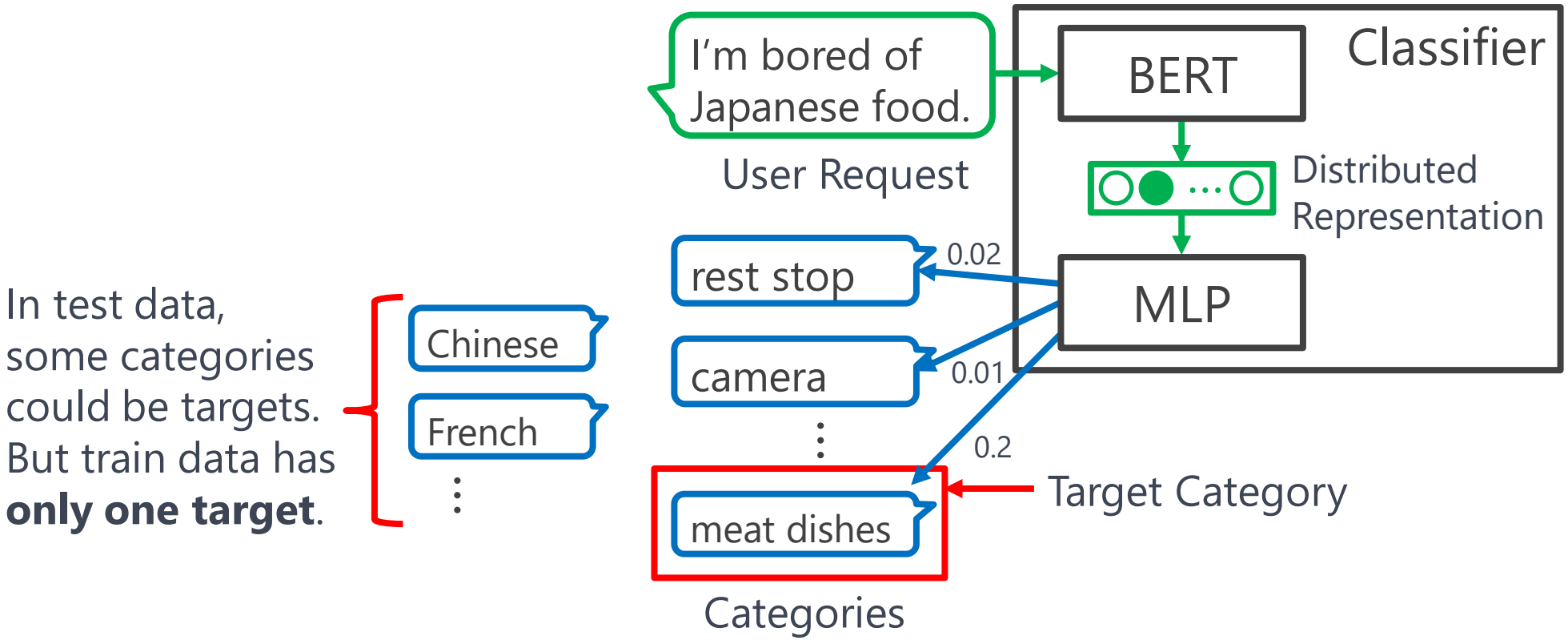
Std is 7.84; # added categories varies greatly for each user request.

Actions related to the same role are annotated in functions of "**spot search**" and "**restaurant search**."

One of the actions near the right-most column is identified as thoughtful for many requests.

The action category is "**browser**," which is expressed in the form of "Shall I display the information about XX?"

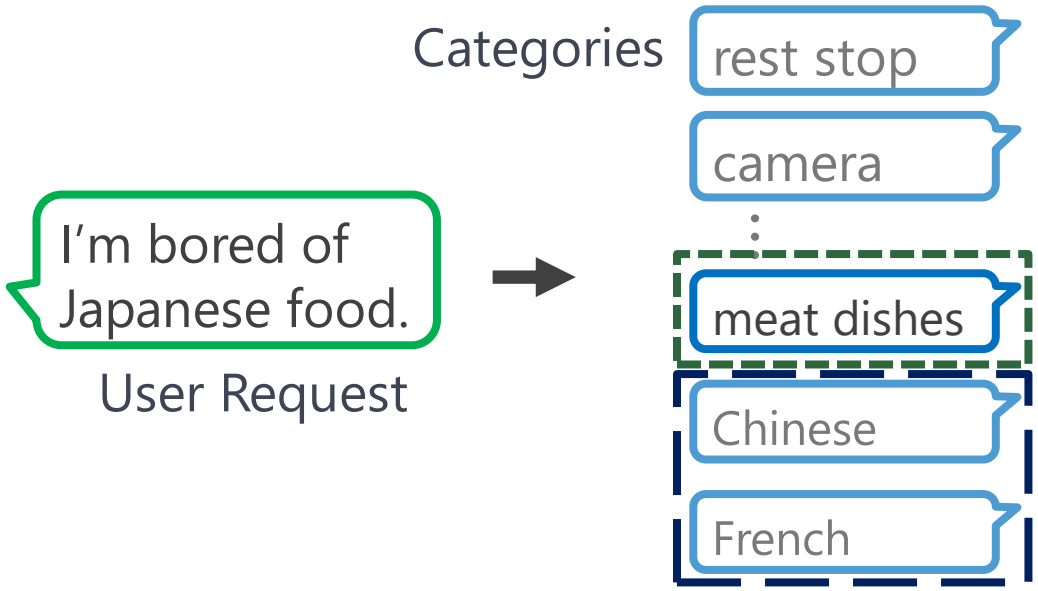
User Request Classifier



Classify user requests to each target category.

- BERT converts the requests to distributed representations.
- MLP calculates the probability value for each category.
- Category with the highest probability is the predicted category.

Training Model with Uncomplete Labeled Data



In **PN learning**, only meat dishes is treated as positive (labeled). Other categories are unlabeled; Some of them could be positive. We want to treat Chinese and French as positive with **PU learning**.

Positive Negative (PN) learning

Treats all combinations of unlabeled user requests and system actions as negative.

In our corpus, about 10% of these combinations should be treated as positive examples.

Positive Unlabeled (PU) learning

PU assumes that some of the data are labeled as positive and the rest are not.

Treats some unlabeled data as positive based on some kind of features.

PU Learning Based on Label Propagation

Label propagation based on nearest neighbor **(PU, nearest)** [Cevikalp et al., 2020]

Propagates labels from the **nearest neighbor** on the distributed representation space.

e.g. shrine -> meat dishes

Proposed Label propagation based on mean vector **(PU, mean)**

Original label propagation is sensitive for outliers.

e.g. propagation from shrine to meat dishes is usually wrong.

Propagates labels according to their distance from **mean vectors** of each category.

Please refer to our paper for the detailed equations!

Experimental Settings

Used pretrained Japanese BERT models [Shibata et al., 2019]

Used BASE and LARGE [Delvin et al., 2019] BERT models.

Pre-trained the models by PN learning before we applied PU learning.

Evaluation metrics

- Accuracy
- R@5: Percentage of a target category ranking within the top 5
- Mean Reciprocal Rank (0 < MRR <= 1):

$$MRR = \frac{1}{|U|} \sum_{u \in U} \frac{1}{rank_u}$$

$|U|$: User requests in test data

$rank_u$: Rank of target category

Classification Results

* means that $p < 0.01$.

Model	Accuracy (%)	R@5 (%)	MRR
BASE (PN)	88.33 (+-0.92)	97.99 (+-0.25)	0.9255 (+-0.0056)
BASE (PU, nearest)	88.29 (+-0.96)	97.81 (+-0.27)	0.9245 (+-0.0056)
BASE (PU, mean)	***89.37** (+-0.78)	97.85 (+-0.26)	***0.9305** (+-0.0050)
LARGE (PN)	89.16 (+-0.57)	98.08 (+-0.22)	0.9316 (+-0.0032)
LARGE (PU, nearest)	89.06 (+-0.66)	98.01 (+-0.24)	0.9295 (+-0.0036)
LARGE (PU, mean)	***90.13** (+-0.51)	98.11 (+-0.27)	***0.9354** (+-0.0035)

(PU, Mean) achieved significant improvement over the baseline method (PN) on accuracy and MRR.

No improvement on R@5 because correct actions are already included in the top five.

Label Propagation Performance

Model	Precision (%)	Recall (%)	F1
BASE	78.06 (+-3.35)	8.53 (+-1.31)	0.1533 (+-0.0206)
LARGE	**79.27** (+-4.43)	7.91 (+-1.10)	0.1435 (+-0.0172)

Evaluated the label propagation performance in the proposed method (PU, mean) on the test data.

The higher the precision of the label propagation, the higher the performance of the model.

Label propagation is able to add thoughtful action categories as positive examples with high precision.

There is still room for **improvement on their recalls**.

Conclusion

Summary

Collected a corpus consists of ambiguous user requests and thoughtful actions.

Constructed test data as a multi-class classification problem.

Developed user request classifiers using BERT.

Proposed PU learning method achieved high accuracy.

Future Work

Updating the classifier by improving the label propagation performance

Investigating the features of user requests that are difficult to classify