Effective Dialogue-Context Retriever for Long-Term Open-Domain Conversation

Meguru Takasaki, Naoki Yoshinaga and Masashi Toyoda

Abstract With the spread of smart speakers, open-domain dialogue systems are expected to have long-term communication with their users. In such circumstances, the dialogue systems need to generate responses by taking relevant past conversations into account. However, due to the length limitation of input in neural dialogue systems and the acceptable latency of dialogue systems, we cannot take all the dialogue histories into account. In this study, we propose a task-specific retriever for effectively extracting a core fragment of dialogue histories that are useful to reply to a given dialogue context. We experimentally confirm the advantage of our retriever against the existing session-based retriever on a GPT-2-based dialogue system trained with a large-scale Twitter dataset.

1 Introduction

Dialogue systems are becoming our daily conversation partners since they become available as virtual assistants on smartphones (*e.g.*, Apple Siri) and smart speakers (*e.g.*, Amazon Echo). These virtual assistants are expected to not only answer voice-based requests but also have open-domain chit-chat with the users; improving the chit-chat ability is the key to increasing user engagement (Bickmore and Picard, 2005). Although online conversation logs on microblogs facilitate research on data-driven open-domain dialogue systems (Ritter et al, 2011; Wang et al, 2013; Al-Rfou et al, 2016), their conversation ability is limited even using neural generation models. This is because the task is modeled to mimic human responses given a limited

Meguru Takasaki
The University of Tokyo,
e-mail: takasa-m@tkl.iis.u-tokyo.ac.jp

Naoki Yoshinaga and Masashi Toyoda
Institute of Industrial Science, the University of Tokyo,
e-mail: {ynaga,toyoda}@iis.u-tokyo.ac.jp

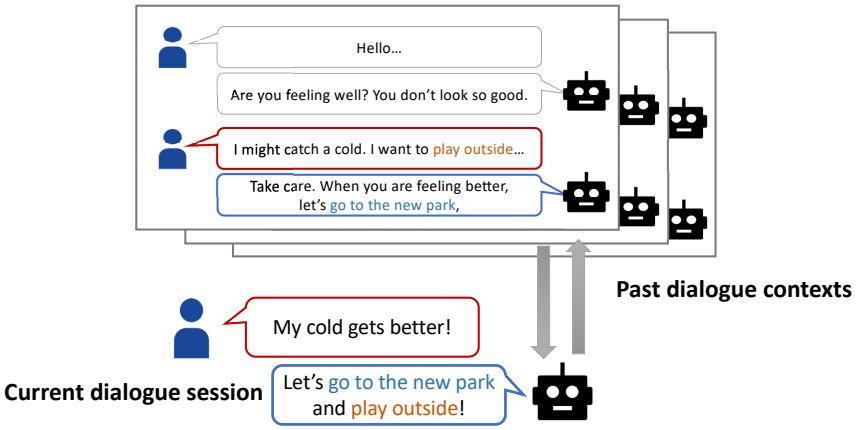

Fig. 1 Response generation by using dialogue contexts in the past sessions. In the past sessions, some dialogue contexts could be relevant to the current dialogue contexts (red) and following utterances could be useful for generation responses (blue), while the others are irrelevant (gray).

length of dialogue contexts as inputs, and effective training is inherently difficult due to the degree of freedom on responses (Sato et al, 2017); even a human cannot continue a conversation for many turns with others s/he have never met before.

Assuming long-term conversation between a deployed dialogue system and their specific users, recent studies utilized the past conversational sessions to generate responses personalized to the user (Xu et al, 2022a,b; Bae et al, 2022). However, due to the length limitation of inputs for over-parametrized Transformer-based dialogue systems (Adiwardana et al, 2020; Roller et al, 2021), naively retrieving relevant sessions does not improve the quality of generated responses (Xu et al, 2022a). They therefore resort to dialogue summaries (Xu et al, 2022a) and accumulated persona information (Xu et al, 2022b; Bae et al, 2022) instead of raw dialogue contexts. Although such compressed information works effectively, extra supervision required to obtain such information cancels out the merits of data-driven dialogue modeling that benefits from massive raw conversation logs.

In this study, we propose effective task-specific retrievers for long-term conversation (Fig. 1) and evaluate the proposed retrievers on a nine-year worth of large-scale Twitter conversation datasets. Our retriever extracts core fragments of dialogue histories that are useful to reply to given dialogue contexts in the current session, requiring no extra supervision for training. We explore effective units of queries to match with keys given to fragments of dialogue contexts in the past sessions. Both queries and keys are embedded using Sentence-BERT (Reimers and Gurevych, 2019) which is trained to minimize the distance between dialogue contexts in the same session.

We trained our proposed models on a long-term conversational Twitter dataset (§ 3), which contains more than ten dialogue sessions between each pair of specific users. Experimental results on this dataset indicate that our task-specific retriever performs better than the existing session-based retriever on a GPT-2-based dialogue system in terms of automatic metrics and human judgments on generated responses.

2 Related work

In this section, we first review previous studies on retrieved-guided response generation for dialogue systems. Then, we compare our work with other dialogue systems for long-term open-domain conversation. Finally, we review existing studies on retrieval-augmented models for other language tasks.

2.1 Retrieval-guided Response Generation

Some studies retrieved and utilized responses in the conversation logs to generate informative and adequate responses. Assuming that similar conversations often occur in closed-domain conversations, Pandey et al (2018) leveraged responses to similar dialogue contexts in the closed-domain conversation logs (Lowe et al, 2015) to generate responses. Cai et al (2019) created response skeletons from retrieved responses to generate responses. Wu et al (2019) edited extracted responses by taking the difference between their contexts and the given context into account.

Whereas these studies retrieve responses from conversational logs spoken by various users to generate user-agnostic responses, our study utilizes past dialogue sessions given by the target user to adjust responses to that speaker. Our method leverages a pre-trained generative model to generate fluent responses.

2.2 Long-term Open-domain Conversation

Recent studies developed dialogue datasets and response generation methods for long-term conversation. Xu et al (2022a) built a long-term conversation dataset, Multi-Session Chat (MSC), and proposed a dialogue system that summarizes the past dialogue sessions and injects the summaries into the response generator. They have reported that retrieval-augmented generation (Lewis et al, 2020; Izacard and Grave, 2021) were not as effective as their dialogue summary-based method when whole dialogue sessions were retrieved. Xu et al (2022b) generated chat responses by referring to accumulated persona information. Bae et al (2022) built a long-term conversation dataset, CareCall$_{mem}$, in which personal information is updated more frequently than MSC. They also proposed to keep users' summaries updated during a conversation, enabling to track memory changes across multiple sessions.

Although these methods could utilize information conversed in the past sessions, they require manual supervision to induce summaries or persona. Meanwhile, our dialogue-context retriever does not require manual supervision for training. While the session-based retriever (Xu et al, 2022a) will inject useless contexts into the generator, our retriever retrieve dialogue contexts in a smaller unit than a session.

To evaluate our method, we newly create a multi-session dialogue dataset from human-human dialogues on Twitter. Our dataset includes a wide range of topics,

		Train	Dev.	Test
All sessions	Number of episodes	60,000	1778	2682
	Collection periods	2011 – 2017	2018	2019
	Number of predicted utterances	150,747	4666	7113
	Average number of tokens per episodes	2,593.65	2,654.47	2,561.63
	(the ratio over 1024 tokens) (%)	98.99	99.44	98.66
Current sessions	Average number of turns per one session	6.65	6.89	6.89
	Average number of tokens per one session	146.37	151.51	146.66
Past sessions	Average number of turns per one session	6.86	7.17	7.04
	Average number of tokens per one session	153.68	162.41	155.93
	Average number of sessions	15.92	15.41	15.49

Table 1 Details of our multi-session Twitter dialogue datasets.

while the MSC dataset has topics mostly limited to given profiles and CareCall$_{mem}$ has pre-defined 89 topics. Our dataset contains more than twice and four times as many tokens per episode as MSC and CareCall$_{mem}$ datasets, respectively, and most of the episodes are longer than the max length of the pre-trained model (1024).

2.3 Retrieval-augmented Models for Other Language Tasks

To perform other language tasks than response generation, several studies dynamically retrieve knowledge for a given input, using keys, queries, and values tailored for the target tasks. Language models using external knowledge have been proposed to perform open-domain question answering and language modeling (Guu et al, 2020; Lewis et al, 2020; Izacard and Grave, 2021). They created queries from the whole input or its fixed-length chunks. Shinzato et al (2022) used attributes as keys to retrieve their possible values from the training data to perform product attribute-value extraction. Wang et al (2022) designed keys and values to retrieve knowledge from the training data for summarization, language modeling, machine translation, and question-answering. Nishida et al (2023) leveraged unconfident entities in the input as queries to perform self-adaptive named entity recognition.

These studies emphasize that properly designing queries, keys, and values for the target task is essential to improve task performance. In this study, we explore various methods for creating queries, keys, and values to effectively retrieve useful past dialogue contexts for long-term open-domain conversation.

3 Multi-session Twitter Dialogue Datasets

In this study, we built a Japanese long-term open-domain dialogue dataset from conversation logs on Twitter. The statistics of the dataset are given in Table 1. Specifi-

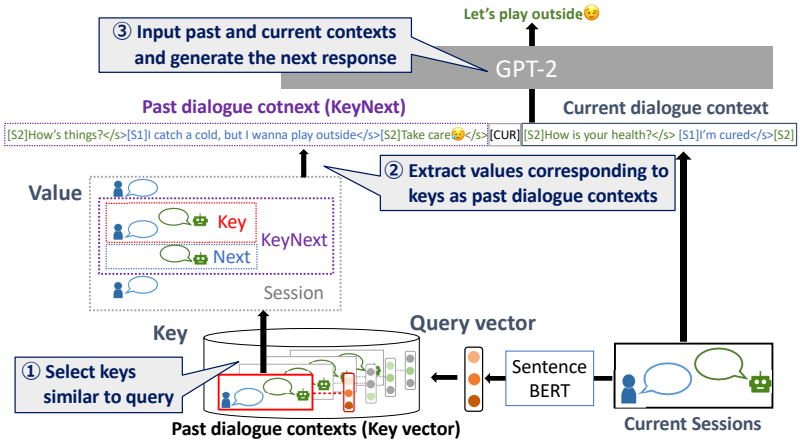

Fig. 2 Overview of our dialogue-context retriever for long-term open-domain conversation.

cally, to obtain the multi-session dialogue dataset, we leveraged our Twitter archive[1] that were retrieved by using the Twitter API.[2] We release tweet IDs[3] of our datasets; researchers can rebuild our datasets using the Twitter API.

We regarded one reply tree as one dialog session and only used dialog sessions consisting of utterances alternately posted by two specific users; we referred to dialogue sessions retrieved for two specific users as 'episodes' in this paper. We split the obtained episodes into training, development, and test sets so that the speakers do not overlap with each other. Also, we removed dialogues containing URLs, images, and posts tweeted by bots. In order to exclude too short or long dialogues, we used only episodes with 11-25 sessions which consist of 5-30 turns. In experiments, we assume that these conversations are performed between a dialogue system and its user, and dialogue systems are trained to generate responses spoken by either of the two Twitter users. In training and testing our dialogue-context retriever for dialogue systems, assuming a Twitter user who starts the conversation in the final session as a user and the other user as a dialogue system, the dialogue systems are requested to generate responses for $2n$-th ($n > 0$) user uttterances in the final session.

4 Proposed method

In this section, we propose a task-specific retriever for long-term open-domain response generation (Fig. 2). Our retriever is trained solely on a multi-session dia-

[1] Starting from 26 popular Japanese users in Mar. 2011, their timelines (recent tweets) have been continuously collected by user_timeline API, while the user set has been iteratively expanded to those who were mentioned or whose tweets were reposted by already seen users.

[2] `https://developer.twitter.com/en/docs/twitter-api`

[3] `https://www.tkl.iis.u-tokyo.ac.jp/~takasa-m/iwsds2023.html`

logue dataset without any manual supervision, and extracts various-sized dialogue contexts in the past sessions. Finally, the retrieved dialogue contexts are injected along with the dialogue contexts in the current session into the Transformer decoder, which is fine-tuned on the top of the pre-trained GPT-2 (Radford et al, 2019), as is shown in Fig. 2. In contrast to RAG (Lewis et al, 2020) and FiD (Izacard and Grave, 2021), our retrieval-augmented dialogue system does not need an encoder for integrating retrieved values. Therefore, this method can be easily utilized not only by encoder-decoder models but also by decoder-only models like GPT-2 which is commonly used in dialogue modeling (Zhang et al, 2020).

4.1 Past Dialogue-Context Retriever

Our dialogue-context retriever retrieves useful dialogue contexts in the past sessions for a given dialogue context in the current session (§ 4.1.1) by using Sentence-BERT (§ 4.1.2). In what follows, we first explain how to design keys, queries, and values to retrieve relevant dialogue contexts, and then explain how to embed dialogue contexts for nearest-neighbor search.

4.1.1 Extracting dialogue context

To begin with, we need to create queries and keys for retrieving relevant dialogue contexts as values; queries are generated from the current context, and keys are made from utterances in the past sessions between the same pairs of speakers (assuming the target user and the dialogue system). The queries and keys are composed of the same number of utterances (1 to 3), end with the same target speaker, and are vectorized to retrieve values associated with keys according to similarity scores between keys and values (§ 4.1.2) because it will be difficult to compute similarities between dialogue contexts with different lengths. We explore various types and lengths (the number of utterances) of queries and keys in the following experiments.

We consider that the existing session-based keys, queries, and values (Xu et al, 2022a) are not necessarily suitable for retrieving past dialogue contexts in the response generation task, since the conversation topics can change during a single dialogue session. Therefore, we consider small fragments of dialogue contexts in the past sessions as values for dialogue context retrieval. In experiments, we compare the following four types of values associated with keys, as shown in Fig. 2:

Session Inject whole sessions with the selected keys to the response generator (Xu et al, 2022a). The retrieved sessions may contain irrelevant information, which may occupy the limited input length of the response generator.

Key Inject the selected keys as values to the response generator. Similar keys to the queries will help the generator to understand the current dialogue contexts.

Next Inject the next utterance of the selected key to the response generator. Next utterance help the generator to capture common utterances for the current context (Pandey et al, 2018).

Key+Next Inject both the selected key and its next utterance into the response generator.

We feed as many values of the similar keys as possible to the generator; we choose keys from the ones with the highest similarity scores one by one, until the total number of tokens (including those tokens in the given dialogue contexts in the current session) exceeds the length limit of the generator inputs (256 or 1024).

4.1.2 Embedding Dialogue Contexts using Sentence-BERT

We next vectorize queries generated from current dialogue contexts and keys generated from past contexts by using Sentence-BERT (Reimers and Gurevych, 2019). Sentence-BERT is fine-tuned BERT (Devlin et al, 2019) for semantic similarity search, consisting of the original BERT model and the pooling layer.

Sentence-BERT requires a training dataset consisting of sentence triplets in order to learn the distance between sentences (here, dialogue contexts). In this work, we want to make the distance between dialogue contexts that are relevant and could be in the same session closer, and make the distance between dialogue contexts that are irrelevant and will not appear in the same session more distant. Therefore, we create the dataset with the anchor and the positive dialogue contexts coming from conversation logs in the same session, and the negative dialogue contexts sampled from the other sessions between the same user pair. All dialogue contexts in the same triplet consist of the same number of utterances (1 to 3) that end with the utterances given by the same speaker.

We use a triplet loss (Reimers and Gurevych, 2019) as the objective function. The triplet loss makes the distance between the anchor a and the positive example p closer, and makes the distance between a and the negative example n more distant:

$$L_{triplet} = max(|s_a - s_p| - |s_a - s_n| + \varepsilon, 0), \tag{1}$$

where s_a, s_p and s_n are the embeddings of the anchor, the positive example, and the negative example, respectively. The negative example is at least ε further away from the anchor than the positive example.

4.2 Response Generator Guided by Retrieved Past Contexts

After retrieving relevant dialogue contexts from the past sessions, we concatenate the retrieved values and the given dialogue contexts in the current session to feed them into the generator. The generator is fine-tuned from pre-trained GPT-2 (Radford et al, 2019) by using cross-entropy loss for each reference response. Given

current context $\boldsymbol{x}$, the selected n values $\boldsymbol{v} = (v_1, ..., v_n)$, and reference response $\boldsymbol{y}$ (consisting of $|\boldsymbol{y}|$ tokens and using $y_{1:t-1} = y_1, y_2, \ldots, y_{t-1}$), the loss is as follows:

$$L_{CE} = -\sum_{t=1}^{|\boldsymbol{y}|} \log p(y_t | \boldsymbol{v}, \boldsymbol{x}, y_{1:t-1}) \tag{2}$$

Speaker tokens `[S1]` or `[S2]` are added at the beginning of each utterance in order to indicate the speaker of the utterance consistently. Also, we end each utterance with an "end of sentence" token `</s>`. We separate selected values with "end of one history" tokens `[EOH]`, and put a `[CUR]` token to indicate the boundary between the end of the selected final value and the current context.

5 Experiments

We evaluated our proposed retriever on the long-term open-domain response generation task using a massive Twitter dataset (§ 3) that consists of multiple dialog sessions between two human speakers. We fine-tuned GPT-2-based dialogue models with our dialogue-context retriever on the Twitter dataset, and then evaluate the performance in terms of automatic metrics and human judgments.

5.1 Models

We fine-tuned a pre-trained GPT-2 (Radford et al, 2019)[4] with our dialogue context retriever on our Twitter dialogue datasets, and then compared the resulting models with two baselines: GPT-2-based dialogue models fine-tuned without past contexts and with most recent past sessions. We implemented all models using PyTorch 1.10.2 (Paszke et al, 2019) and Transformers 4.20.0 (Wolf et al, 2020). We created training (and development) examples for Sentence-BERT (§ 4.1.2) from individual episodes in training (and development) set, which are sets of dialogue sessions between specific two users. We used a pre-trained BERT[5] with an additional pooling layer as a base of our Sentence-BERT. We fine-tuned this model for five epochs and chose the model that achieved the best accuracy on the development set.

We trained all dialogue models for at most 5 epochs with early stopping regularization whose patience is one epoch, and used the best model for evaluation in terms of perplexity on the development set. We varied the length of queries and keys (1 to 3 utterances and their combination), the types of value (Session, Key, Next, and KeyNext) as described in § 4.1.1, and the maximum length of input and output. All models were fine-tuned with a maximum of two 24GB GPUs (Quadro

[4] `https://huggingface.co/rinna/japanese-gpt2-small`

[5] `https://huggingface.co/cl-tohoku/bert-base-japanese-v2`

Model	Values	PPL (↓)	BLEU-2/3 (↑)	ROUGE-L (↑)	BERTScore (↑)	DIST-1/2 (↑)	# values
No past contexts		49.71	1.21/0.56	8.17	59.08	7.53/22.39	-
Truncated by 256							
Baseline	Session	45.53	1.99/0.82	10.24	60.70	5.03/19.09	1.70
Ours	Session	45.62	1.89/0.76	10.26	60.84	5.07/19.63	1.61
	Key	47.34	1.75/0.76	9.69	60.02	4.97/18.19	7.69
	Next	**44.21**	**2.14**/**0.88**	**10.65**	**60.95**	5.31/20.46	8.03
	KeyNext	44.91	2.02/0.84	10.26	60.76	**5.60**/**21.41**	4.54
Truncated by 1024							
Baseline	Session	42.94	2.02/0.84	10.12	61.32	5.47/22.67	7.30
Ours	Session	43.19	1.94/0.83	9.83	61.06	5.73/22.84	6.68
	Key	46.88	1.57/0.63	9.15	59.92	5.01/19.59	40.98
	Next	42.11	1.71/0.79	9.12	60.21	5.98/23.36	39.44
	KeyNext	**41.78**	**2.10**/**0.86**	**10.42**	**61.59**	**6.20**/**25.39**	25.25

Table 2 Automatic Evaluation of GPT-2-based dialogue systems with and without our dialogue-context retriever on all turns (The keys and queries of our retriever are single utterances).

P6000) with eight batch size per GPU. In order to save memory, we used DeepSpeed (Rasley et al, 2020) library with Stage-3 and half precision floating point (FP16). We used AdamW (Loshchilov and Hutter, 2019) optimizer with $\beta 1 = 0.9$, $\beta 2 = 0.999$, $\varepsilon = 1\text{e}{-10}$. The fine-tuning takes up to 16 hours before it early stops.[6]

5.2 Automatic Evaluation

We first evaluate responses generated by the dialogue systems by using automatic metrics. Specifically, we measured perplexity, BLEU-2/3 (Papineni et al, 2002), F_1 of ROUGE-L (Lin and Hovy, 2003), and DISTINCT-1/2 (Zhao et al, 2017). We used MeCab[7] with UniDic 2.1.2 to tokenize the generated responses. We also measured F_1 of BERTScore (Zhang* et al, 2020), which has a high correlation with human judgements (Yeh et al, 2021) on the Twitter-based dataset (Hori and Hori, 2017), using a pre-trained Japanese RoBERTa (Liu et al, 2019).[8] We also calibrated the number of values injected into the model after truncation.

Table 2 shows the results of the automatic evaluation. The proposed methods use single utterance-based keys and queries for retrieving past dialogue contexts. We observe a clear improvement in perplexity and DISTINCT-1/2 when inputting utter-

[6] The total length of input and output must be at most 1024 (or 256) tokens in GPT-2. To make the fine-tuned GPT-2 generate responses of appropriate length, we set the maximum length of output to 48 so that it covers 95% of reference responses and excluded examples with longer responses from training. We then use the remaining tokens for the current and retrieved dialogue contexts.

[7] `https://taku910.github.io/mecab/`

[8] `https://huggingface.co/nlp-waseda/roberta-base-japanese`

Model	Values	PPL (↓)	BLEU-2/3 (↑)	ROUGE-L (↑)	BERTScore (↑)	DIST-1/2 (↑)	# values
No past contexts		48.18	1.15/0.50	7.76	58.70	10.45/22.66	-
Truncated by 256							
Baseline	Session	41.46	1.84/0.82	9.70	59.88	7.79/23.57	2.01
Ours	Session	41.54	1.70/0.68	9.61	59.93	7.83/23.75	1.85
	Key	44.56	1.62/0.73	8.90	58.58	7.00/19.92	9.25
	Next	**39.25**	**2.07**/**0.89**	**10.27**	**60.29**	8.49/25.62	9.91
	KeyNext	40.49	1.79/0.75	9.68	59.86	**8.71**/**26.19**	5.41
Truncated by 1024							
Baseline	Session	38.26	1.70/0.59	9.60	60.46	8.86/29.01	7.68
Ours	Session	38.49	1.86/**0.82**	9.13	60.20	9.22/28.86	7.09
	Key	43.87	1.19/0.44	8.01	58.33	6.75/20.00	41.61
	Next	**36.73**	1.53/0.65	8.55	59.46	9.80/30.30	39.82
	KeyNext	37.02	**1.99**/0.78	**10.06**	**60.96**	**9.97**/**32.81**	25.99

Table 3 Automatic Evaluation of the responses to the first utterance in the session (The length of keys and queries of our retriever is one utterance).

ances next to extracted similar past contexts. The Next retriever performs best when input text is truncated by 256 tokens, and the KeyNext retriever performs the best when truncated by 1024 tokens. On the other hand, the improvement of reference-based metrics (BLEU-2/3, ROUGE-L, and BERTScore) is small. Therefore, we perform paired bootstrap resampling (Koehn, 2004) to confirm the significance. The dataset size for random sampling is as same as the original test dataset, and the number of resampling is 1000 times. We compare our best methods (the Next when truncated by 256 tokens and the KeyNext when truncated by 1024 tokens) with baselines using the same length of truncation. As a result, our best-performing methods are significantly better than the baselines ($p < 0.05$) for all the metrics other than BLEU-2 and BLEU-3 for KeyNext methods. Meanwhile, the Key and Session retrievers perform worse than the baselines with most recent past sessions. This result suggests that keys and their previous utterances are less useful than their next utterance. Also, it suggests that inputting past information in chronological order could be more useful than inputting them in similarity-based order.

Evaluation on the first response Table 3 indicates the results of the evaluation on generating responses to the first utterance in the session. In this situation, the baseline model truncated by 1024 performs worse than the baseline truncated by 256 in terms of BLEU-2/3 and ROUGE-L. This is because that prediction for the second utterance cannot utilize enough context in the current session and are likely to rely on past contexts. Therefore, noisy session-based past contexts deteriorate the performance of baselines when truncated by 1024. Our Session retriever, however, improves its performance when the truncation length increased from 256 to 1024. This is because input past contexts are at least similar to the current context and would be less noisy. Meanwhile, our Next retriever performs best when truncated by 256 tokens, and the KeyNext retriever performs the best when truncated by 1024

Sorting methods	PPL (↓)	BLEU-2/3 (↑)	ROUGE-L (↑)	BERTScore (↑)	DIST-1/2 (↑)
Time	**42.19**	**1.97**/**0.76**	**10.15**	**61.30**	**5.63**/**23.97**
Similarity	42.36	1.94/0.81	10.10	61.22	5.45/23.23

Table 4 Comparison of the order of inputting past dialogue contexts.

Query types	PPL (↓)	BLEU-2/3 (↑)	ROUGE-L (↑)	BERTScore (↑)	DIST-1/2 (↑)	# values
Truncated by 256						
1 utt.	**44.91**	**2.02**/0.84	10.26	60.76	5.60/**21.41**	4.54
max 2 utts.	45.26	1.84/0.79	9.89	60.65	5.39/20.25	3.82
max 3 utts.	45.09	1.88/0.74	**10.28**	**60.81**	5.12/19.85	3.43
1+2+3 utts.	45.44	1.93/**0.85**	9.86	60.54	**5.71**/21.05	4.51
Truncated by 1024						
1 utt.	**41.78**	2.10/0.86	**10.42**	**61.59**	**6.20**/**25.39**	25.25
max 2 utts.	42.52	2.09/0.86	10.39	61.36	5.52/22.92	20.07
max 3 utts.	42.11	2.12/**0.87**	10.28	61.33	5.75/23.48	17.28
1+2+3 utts.	41.94	**2.15**/0.82	10.62	61.55	5.81/24.22	25.05

Table 5 Comparison of query lengths (All results come from our proposed method using KeyNext values and the limit of input length is 1024).

tokens. This suggests that these methods can utilize past useful information and be less likely to be affected by past noisy contexts.

Impact of ordering retrieved past contexts Table 4 shows a comparison of our KeyNext retriever with a truncation length of 1024 when sorting the retrieved past contexts in various orders. In our methods, extracted past contexts are sorted by similarity. However, this method disregards the time information of past contexts and models cannot consider the chronological relationships between the retrieved past contexts. We thus train our models with different ordering of past contexts. In order not to truncate different past contexts between the two methods, we use as many values as possible so that they do not exceed the models' maximum input length. These models use keys and queries consisting of single utterances. We can observe that the model sorting past contexts by time slightly outperforms the model sorting past contexts by similarity. It indicates that past dialogue contexts should be input in chronological order.

Impact of length of queries and keys Table 5 shows the results of comparing different lengths of queries and keys and their combinations. 1, 2 and 3 utt. means the models using queries and keys whose length is at most one utterance, two utterances, and three utterances, respectively. 1+2+3 utts. refers to the models that extract variable-length queries from the current context and match them with keys with the same length. From the table, we can observe that the dialogue system using single utterances as queries and keys performed the best. This suggests that it is difficult to extract useful past values when using long queries and keys. The number of finally

Model	Coherence	Contextual consistency	Humanness
Baseline	−0.219	−0.0246	0.127
Ours (KeyNext)	**−0.107**	**0.0475**	**0.175**

Table 6 Human Evaluation of dialogue systems with a truncation length of 1024: average standardized scores per annotator.

injected values of 1+2+3 utts. is almost the same as that of 1 utts., suggesting that most selected keys consist of one utterance because they are likely to get higher similarity scores than longer keys.

5.3 Human Evaluation

For human evaluation, we ask three annotators to score responses generated by the baseline with most recent past contexts and the best-performing proposed model with the KeyNext retriever for the length truncation of 1024 in Table 2. Considering that the average number of sessions is 15.49, we sampled ten episodes which contain 15-17 sessions from the test dataset, and annotators evaluated generated responses in the last three sessions. The three annotators ultimately evaluated 116 system responses generated by each dialogue model in terms of the following metrics with a rating scale of 0 to 100; the resulting scores are then standardized per annotator (Ji et al, 2022).

Coherence The model understood the current dialogue context and generated a response coherently.

Contextual consistency The model generated a response that was consistent with the past dialogue sessions.

Humanness This model generated a response like a human.

Table 6 shows the results of the human evaluations. Contextual consistency metric indicates that our method has a great advantage of understanding long-term dialogue context and generating consistent responses (p-value for paired-sample t-test is $0.039 < 0.05$). On the other hand, p-values of coherence and humanness are not small (0.105 and 0.111). These results indicate that our KeyNext retriever significantly improved the contextual consistency while keeping the response quality in terms of coherence and humanness.

5.4 Examples

Fig. 3 and Fig. 4 show responses generated by baseline and our proposed method (KeyNext, 1 utt., truncated by 256). In the current dialogue in Fig. 3, the user talks about harsh weather in Hokkaido. The baseline uses the most recent past dialogue

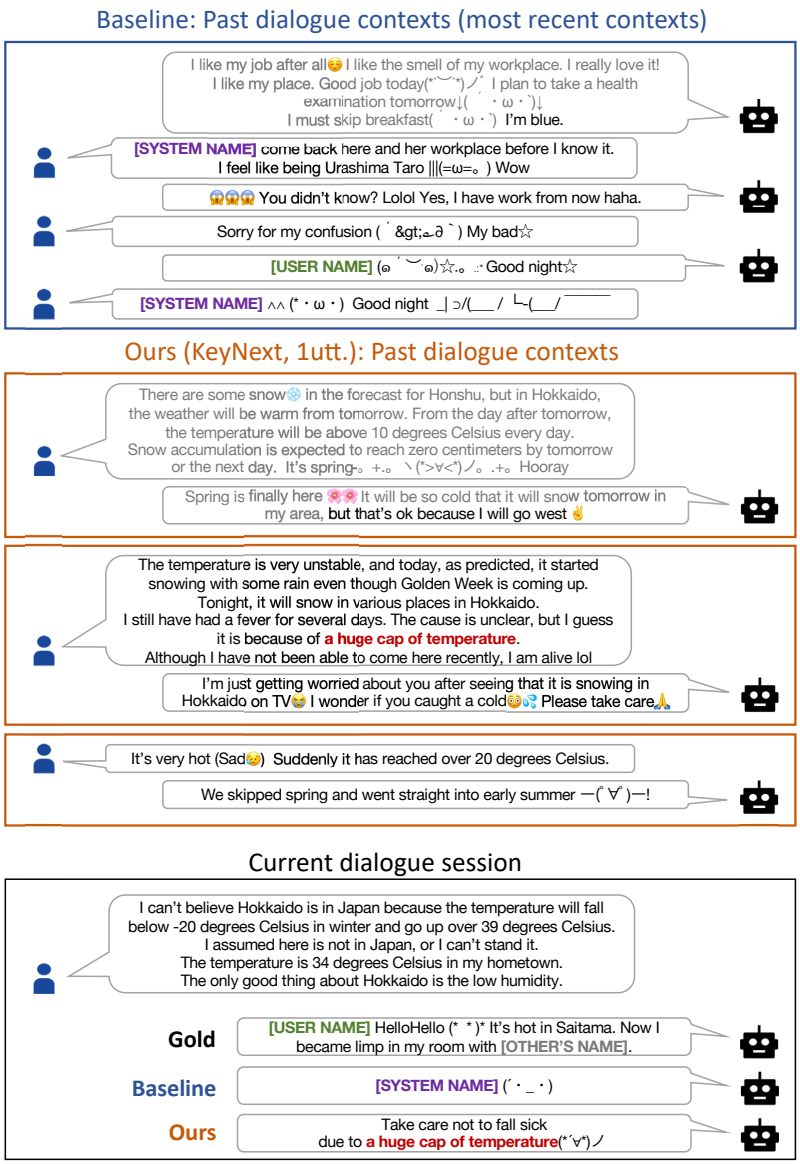

Fig. 3 Example of responses generated by our method (Glay text is truncated when the max input length is 256 tokens); useful past contexts are retrieved and utilized. All texts are translated into English.

contexts about the chatbot-side speaker's job, which is irrelevant to the current session. The response generated by this baseline misunderstood the user names. On the other hand, our method retrieved past dialogue contexts about the weather in Hokkaido, and utilized its information to generate the response. In the dialogue in

Fig. 4 Example of responses generated by our method (Glay text is truncated when the max input length is 256 tokens); useful past contexts are retrieved but not utilized. All texts are translated into English.

Fig. 4, although our method retrieved the past dialogue context which is strongly related to the current dialogue context (about Boba tea, which was a popular beverage in Japan), it did not effectively utilize the contexts to generate responses; it referred to the user name appearing in the other dialogue contexts.

6 Conclusions

We proposed task-specific retrievers for long-term open-domain response generation. Our retriever extracts core fragments from the past context in smaller units than session-based retrievers, which is expected to inject more useful information into the model for response generation. Also, our retriever is trained by using examples automatically generated from the dialogue dataset, and does not require manual supervisioon. Experiments on our long-term Twitter dialogue dataset confirmed that our retriever could outperform the prior session-based retriever in terms of both automatic metrics and human judgments.

Future work should consider how to input time-series information explicitly because our proposed method cannot grasp state information transitions over time. We should also investigate the efficient and effective architecture for handling long inputs, such as sparse attention (Child et al, 2019; Beltagy et al, 2020) and relative position encodings (Shaw et al, 2018).

Acknowledgements This work was supported by JST CREST Grant Number JPMJCR19A4, Japan, and JSPS KAKENHI Grant Number 21H03445 and 21H03494. We also thank the annotators and the anonymous reviewers for their hard work.

References

Adiwardana D, Luong MT, So DR, Hall J, Fiedel N, Thoppilan R, Yang Z, Kulshreshtha A, Nemade G, Lu Y, Le QV (2020) Towards a human-like open-domain chatbot. CoRR abs/2001.09977

Al-Rfou R, Pickett M, Snaider J, Sung Yh, Strope B, Kurzweil R (2016) Conversational contextual cues: The case of personalization and history for response ranking. CoRR abs/1606.00372

Bae S, Kwak D, Kang S, Lee MY, Kim S, Jeong Y, Kim H, Lee SW, Park W, Sung N (2022) Keep me updated! Memory management in long-term conversations. In: Findings of the Association for Computational Linguistics: EMNLP 2022, pp 3769–3787

Beltagy I, Peters ME, Cohan A (2020) Longformer: The long-document transformer. CoRR abs/2004.05150

Bickmore TW, Picard RW (2005) Establishing and maintaining long-term human-computer relationships. ACM Trans Comput-Hum Interact 12(2):293–327

Cai D, Wang Y, Bi W, Tu Z, Liu X, Shi S (2019) Retrieval-guided dialogue response generation via a matching-to-generation framework. 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), pp 1866–1875

Child R, Gray S, Radford A, Sutskever I (2019) Generating long sequences with sparse transformers. CoRR abs/1904.10509

Devlin J, Chang MW, Lee K, 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), pp 4171–4186

Guu K, Lee K, Tung Z, Pasupat P, Chang MW (2020) REALM: Retrieval-augmented language model pre-training. In: Proceedings of the 37th International Conference on Machine Learning, pp 3929–3938

Hori C, Hori T (2017) End-to-end conversation modeling track in DSTC6. CoRR arXiv:1706.07440

Izacard G, Grave E (2021) Leveraging passage retrieval with generative models for open domain question answering. In: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, pp 874–880

Ji T, Graham Y, Jones G, Lyu C, Liu Q (2022) Achieving reliable human assessment of open-domain dialogue systems. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp 6416–6437

Koehn P (2004) Statistical significance tests for machine translation evaluation. In: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, pp 388–395

Lewis P, Perez E, Piktus A, Petroni F, Karpukhin V, Goyal N, Küttler H, Lewis M, Yih Wt, Rocktäschel T, Riedel S, Kiela D (2020) Retrieval-augmented generation for knowledge-intensive NLP tasks. In: Larochelle H, Ranzato M, Hadsell R, Balcan M, Lin H (eds) Advances in Neural Information Processing Systems, vol 33, pp 9459–9474

Lin CY, Hovy E (2003) Automatic evaluation of summaries using n-gram co-occurrence statistics. In: Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, pp 150–157

Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, Levy O, Lewis M, Zettlemoyer L, Stoyanov V (2019) Roberta: A robustly optimized bert pretraining approach. CoRR abs/1907.11692

Loshchilov I, Hutter F (2019) Decoupled weight decay regularization. In: International Conference on Learning Representations

Lowe R, Pow N, Serban I, Pineau J (2015) The Ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In: Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pp 285–294

Nishida K, Yoshinaga N, Nishida K (2023) Self-adaptive named entity recognition by retrieving unstructured knowledge. In: Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, to appear

Pandey G, Contractor D, Kumar V, Joshi S (2018) Exemplar encoder-decoder for neural conversation generation. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp 1329–1338

Papineni K, Roukos S, Ward T, Zhu WJ (2002) BLEU: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pp 311–318

Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, Lin Z, Gimelshein N, Antiga L, Desmaison A, Kopf A, Yang E, DeVito Z, Raison M, Tejani A, Chilamkurthy S, Steiner B, Fang L, Bai J, Chintala S (2019) PyTorch: An imperative style, high-performance deep learning library. In: Wallach H, Larochelle H, Beygelzimer A, d'Alché-Buc F, Fox E, Garnett R (eds) Advances in Neural Information Processing Systems, vol 32

Radford A, Wu J, Child R, Luan D, Amodei D, Sutskever I, et al (2019) Language models are unsupervised multitask learners. OpenAI blog 1(8):9

Rasley J, Rajbhandari S, Ruwase O, He Y (2020) DeepSpeed: System optimizations enable training deep learning models with over 100 billion parameters. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp 3505–3506

Reimers N, Gurevych I (2019) Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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), pp 3982–3992

Ritter A, Cherry C, Dolan WB (2011) Data-driven response generation in social media. In: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, pp 583–593

Roller S, Dinan E, Goyal N, Ju D, Williamson M, Liu Y, Xu J, Ott M, Smith EM, Boureau YL, Weston J (2021) Recipes for building an open-domain chatbot. In: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, pp 300–325

Sato S, Yoshinaga N, Toyoda M, Kitsuregawa M (2017) Modeling situations in neural chat bots. In: Proceedings of ACL 2017, Student Research Workshop, pp 120–127

Shaw P, Uszkoreit J, Vaswani A (2018) Self-attention with relative position representations. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), pp 464–468

Shinzato K, Yoshinaga N, Xia Y, Chen WT (2022) Simple and effective knowledge-driven query expansion for QA-based product attribute extraction. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pp 227–234

Wang H, Lu Z, Li H, Chen E (2013) A dataset for research on short-text conversations. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pp 935–945

Wang S, Xu Y, Fang Y, Liu Y, Sun S, Xu R, Zhu C, Zeng M (2022) Training data is more valuable than you think: A simple and effective method by retrieving from training data. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp 3170–3179

Wolf T, Debut L, Sanh V, Chaumond J, Delangue C, Moi A, Cistac P, Rault T, Louf R, Funtowicz M, Davison J, Shleifer S, von Platen P, Ma C, Jernite Y, Plu J, Xu C, Le Scao T, Gugger S, Drame M, Lhoest Q, Rush A (2020) Transformers: State-of-the-art natural language processing. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pp 38–45

Wu Y, Wei F, Huang S, Wang Y, Li Z, Zhou M (2019) Response generation by context-aware prototype editing. Proceedings of the AAAI Conference on Artificial Intelligence 33(01):7281–7288

Xu J, Szlam A, Weston J (2022a) Beyond goldfish memory: Long-term open-domain conversation. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp 5180–5197

Xu X, Gou Z, Wu W, Niu ZY, Wu H, Wang H, Wang S (2022b) Long time no see! open-domain conversation with long-term persona memory. In: Findings of the Association for Computational Linguistics: ACL 2022, pp 2639–2650

Yeh YT, Eskenazi M, Mehri S (2021) A comprehensive assessment of dialog evaluation metrics. In: The First Workshop on Evaluations and Assessments of Neural Conversation Systems, pp 15–33

Zhang* T, Kishore* V, Wu* F, Weinberger KQ, Artzi Y (2020) BERTScore: Evaluating text generation with BERT. In: International Conference on Learning Representations

Zhang Y, Sun S, Galley M, Chen YC, Brockett C, Gao X, Gao J, Liu J, Dolan B (2020) DIALOGPT: Large-scale generative pre-training for conversational response generation. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pp 270–278

Zhao T, Zhao R, Eskenazi M (2017) Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp 654–664