Computer Speech and Language 78 (2023) 101460

Contents lists available at ScienceDirect

Computer Speech & Language

journal homepage: www.elsevier.com/locate/csl

Knowledge-grounded dialogue modelling with dialogue-state tracking, domain tracking, and entity extraction

Taesuk Hong [a], Junhee Cho [b], Haeun Yu [b], Youngjoong Ko [b,*], Jungyun Seo [a]

[a] Department of Computer Science and Engineering, Sogang University, 35 Baekbeom-ro Mapo-gu, Seoul, Republic of Korea
[b] Department of Computer Science and Engineering, Sungkyunkwan University, 2066 Seobu-ro Jangan-gu, Suwon, Republic of Korea

ARTICLE INFO

Keywords:
Knowledge-grounded dialogue system
Dialogue-state tracking
Deep learning
Task-oriented dialogue system

ABSTRACT

As knowledge-grounded dialogue systems are attracting intense research interest, technology that facilitates reference to various types of external knowledge as dialogue-system data is also developing. A knowledge-grounded dialogue system must be capable of (1) accurately interpret the conversation content, (2) determining whether external knowledge should be referenced for the current turn, (3) identifying the external knowledge to be referenced, and (4) generating a response. If the referenced external knowledge is in the form of a question–answer pair, this pair is closely related to the domain to which the external knowledge belongs and the entity to which it pertains. This study proposes a knowledge-grounded dialogue system that predicts the domain and entity associated with a question–answer pair referenced by the dialogue system; the system then leverages this information to effectively implement the above three external-knowledge-based capabilities. As the dialogue data associated with external knowledge is diverse, adaptation to the slots and entities of different dialogue datasets is challenging. In this study, the Triple Copy (TripPy) model, which is one of the leading benchmark models for dialogue-state tracking with the Multi-domain Wizard-of-Oz (MultiWOZ) dataset, is further developed to adapt to DSTC10 data for the external knowledge-based dialogue system; hence, dialogue content is effectively interpreted. The developed knowledge-grounded dialogue system incorporates knowledge-seeking turn detection, knowledge selection, and knowledge-grounded response generation models. The model, in DSTC10 dataset, achieves 15.49%p, 12.54%p, and 36.45%p improvements over the baseline, the state-of-the-art model for the previous version of the data (DSTC9 dataset), in terms of the F1 score, Recall@1 score, and BLEU-1 score, respectively. Moreover, in a dialogue-state tracking task for DSTC10 dataset, a 15.41%p improvement in joint goal accuracy score is achieved compared to the TripPy model.

1. Introduction

Knowledge-grounded dialogue systems are currently receiving considerable attention in the study of dialogue systems. In real-world human conversations, various types of knowledge are interchanged between interlocutors. Therefore, design of a conversation model that not only effectively interprets the conversation content, but also appropriately responds using external knowledge, is important. To this end, high-quality conversation datasets such as Multi-domain Wizard-of-Oz (MultiWOZ) 2.1 (Eric et al., 2019) have been made publicly available. In such datasets, each conversation is labelled with corresponding dialogue states that aid

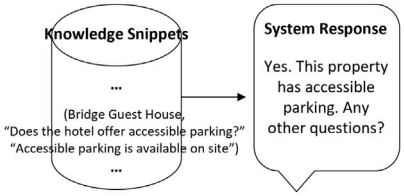
* Corresponding author.
E-mail addresses: lino.taesuk@gmail.com (T. Hong), chojunhee7003@gmail.com (J. Cho), haeun.yu204@gmail.com (H. Yu), yjko@skku.edu, youngjoong.ko@gmail.com (Y. Ko), seojy@sogang.ac.kr (J. Seo).

https://doi.org/10.1016/j.csl.2022.101460
Received 2 December 2021; Received in revised form 16 August 2022; Accepted 7 September 2022
Available online 15 September 2022

Speaker	Utterance
User	I am looking for a moderately priced place to eat, I am also looking to book a room in the bridge guest house hotel.
System	Which dates will you be staying at the Bridge Guest Room house?
User	Before I commit I have a few questions. What area is the hotel located in?
System	The hotel is in the south area.
User	Do they have help for disabled parking?

Domain	Slot	Value
Restaurant	Pricerange	Moderate
Hotel	Name	Bridge Guest House
Hotel	Area	South

1) Dialogue state tracking for dialogue interpretation



2) Knowledge selection for knowledge-seeking turn

Fig. 1. Example of knowledge-grounded conversation. For a given dialogue history, the dialogue state is tracked via the table on the top right. The appropriate external knowledge is selected from among the candidates when the final utterance requests such information.

interpretation of the conversation. In addition, with the growing interest in dialogue-system learning utilizing external knowledge, various corpora such as the Dialogue System Technology Challenge (DSTC9) knowledge-grounded dialogue dataset (Kim et al., 2020), Wizard-Of-Wikipedia (Dinan et al., 2019), and OpenDialKG (Moon et al., 2019) have been released.

Unlike humans who learn information in real time, conversational systems are inevitably limited to the data they learn. Therefore, by using external knowledge other than the learned data, it can generate a response outside the knowledge that the dialogue system has already learned. For a dialogue system to effectively utilize external knowledge, the model must (1) accurately interpret the conversation content, (2) precisely determine whether the current conversation requires the use of external knowledge, and (3) select the most appropriate external knowledge from various types of external knowledge, and (4) properly generate a response. In this study, we develop a dialogue system equipped with the above elements and report the test result on the DSTC (Dialogue System Technology Challenge) 10 knowledge-grounded dialogue system dataset (Kim et al., 2021).

To determine whether external knowledge should be used in a given conversation, it is important for the conversation model to accurately understand the conversation content. External knowledge used in DSTC10 is in the form of a sentence (a sentence of question–answer pair), and there are named entities and domains belonging to each external knowledge. Therefore, if the specific named entity or domain information can be traced in the conversation, the contents of the conversation can be effectively grasped for utilizing external knowledge. In addition, recent conversational research uses a pre-trained language model to enhance understanding of the contents of a conversation. This study proposes a high-performance language model-based dialogue system that can utilize entity and domain information related to external knowledge, and applies it to dialogue state tracking and knowledge-grounded dialogue system tasks in DSTC10 dataset.

Dialogue-state tracking dataset for DSTC10 shares the similar domains and slots as the famous MultiWOZ dataset (Eric et al., 2019), but it differs in values and entity names that can significantly damage the existing dialogue state tracking model. Furthermore, the DSTC10 dataset challenges the task by, for some slots, requesting a label with more than one value. We present a method of effectively enhancing dialogue-state tracking performance by replacing the slots and values of the original MultiWOZ data with those of the DSTC10 dataset. In addition, a data augmentation method for training the dialogue system that can handle a multi-value environment is presented, which differs from the MultiWOZ environment in which only one value can be matched to each existing slot. Finally, we develop useful post-processing techniques to correct the results inferred by the trained model.

In the knowledge-grounded dialogue system task, the capacity of the dialogue system to determine whether external knowledge must be referenced significantly impacts its performance. Fig. 1 presents an example in which the last utterance requires the system to refer to external knowledge. When the system decides to use external knowledge, it must then select the most appropriate external knowledge for response generation, from among the various external knowledge pools, which is challenging. However, recent studies (Karpukhin et al., 2020; He et al., 2021; Guu et al., 2020) have begun to solve this problem by fine-tuning various high-performing pre-trained language models with task-specific data. If information on the entity to which the knowledge is related and the domain to which it belongs is leveraged, this information can be applied to aid external knowledge use by the dialogue system. As an entity related to external knowledge is always required in a knowledge-seeking turn in the DSTC10 dataset (Kim et al., 2021), we develop a entity extraction model using high performing pre-trained model and, in turn, uses the extracted entity to modify the knowledge-seeking turn result. When no knowledge candidate for knowledge selection is found, the knowledge entities belonging to the tracked domain are supplied as candidates to narrow the range of knowledge to be searched.

Our contributions are summarized as follows:

- The proposed dialogue-state tracking model achieves improved performance on the DSTC10 dataset (Kim et al., 2021) through use of data augmentation for value adaptation and multi-value slots. Our model achieves a 15.41%p performance improvement over Triple Copy (TripPy) (Heck et al., 2020), a high-performance model for the MultiWOZ dataset.

- In the knowledge-grounded dialogue system, effective knowledge-seeking turn detection greatly impacts performance. To this end, (1) a high-performing model is developed to extract domains and entities from a conversation, and (2) the performance of both

the knowledge-seeking turn detection model and the knowledge selection model are boosted by leveraging the extracted domains and entities.

2. Related work

2.1. Dialogue-state tracking

Dialogue systems are traditionally designed to engage in conversations with users through dialogue interpretation, dialogue management, and response generation. In the dialogue interpretation stage, a dialogue-state tracking task is performed to map the semantic expressions of the user utterance according to a predetermined slot. The MultiWOZ dataset (Eric et al., 2019) is a dialogue dataset in which users and systems supply continuous utterances about a multi-domain scenario to complete a task. For every turn, the user-utterance dialogue state is automatically annotated through the Wizard-of-Oz dialogue collection process (Dinan et al., 2019). In addition, human experts annotate every system act for the system utterance in the post-processing process. Owing to the sophisticated annotation of this dialogue dataset, MultiWOZ is widely used for dialogue-state tracking model learning for task-oriented dialogue systems. The five highest-ranking models (Dai et al., 2021; Li et al., 2020; Yu et al., 2021; Mehri et al., 2020; Su et al., 2021) in the MultiWOZ leaderboard all use a language model pre-trained with a large corpus. Among them, the representative model is TripPy (Heck et al., 2020). TripPy encodes the conversation content with a pre-trained language model which, in turn, selects a predictive type for each slot of the dialogue state and then selects a value for each type. Currently, performance enhancement was achieved for the three highest-ranked models (Dai et al., 2021; Li et al., 2020; Yu et al., 2021) on the leaderboard using TripPy.

2.2. Knowledge-grounded dialogue systems

In human-to-human conversation, various types of knowledge are combined to expand upon the conversation subject, to question the interlocutor, and to provide answers to questions. To aid construction of a natural dialogue system that reflects these characteristics, various knowledge-based dialogue datasets are being released. Knowledge is presented in various forms, e.g., as structured knowledge as in a DB, text, or document, and as sentence-based knowledge such as a question–answer pair. The MultiWOZ dataset allows response generation for a task-oriented dialogue through reference to knowledge related to a specific value held in a structured DB. Recently, as an advancement from DB referencing, knowledge graphs have been referred to as external knowledge, and data annotated in accordance with the knowledge graph and a pathway to reference this knowledge have also been provided (Moon et al., 2019). Datasets such as ShARC (Saeidi et al., 2018), CoQA (Reddy et al., 2019), FlowQA (Huang et al., 2019), and QAConv (Wu et al., 2021) provide knowledge in document format for reference in response to questions. Further, the dataset provided by Ghazvininejad et al. (2018) , along with the DSTC7 Track-2 (Galley et al., 2019), Wizard-of-Wikipedia (Dinan et al., 2019), and CMU-DoG (Zhou et al., 2018) datasets, relate to casual conversations referring to a document. External knowledge also includes information on a specific individual; among the various datasets of this type, PERSONA-CHAT (Zhang et al., 2018) is popular.

In the DSTC9 Track-1 challenge (Kim et al., 2020), the task was to design a dialogue model that provided the user with required external knowledge outside the general DB, as knowledge obtained through a task-oriented dialogue system. In that case, external knowledge was defined as knowledge snippets in the form of FAQs related to domains and entities. In DSTC9, three subtasks were defined and the corresponding performance was evaluated: knowledge-seeking turn detection, knowledge selection, and knowledge-grounded response generation. The training dataset was an augmented version of MultiWOZ 2.1, in which the crowdsourced augmented component included a turn that referred to external knowledge.

3. Modelling

This section presents our proposed models, which constitute the various components of the developed external knowledge-based dialogue system. Like other aspects of natural language processing, recent trends for conversational systems (Wolf et al., 2019; Zhang et al., 2020; Hosseini-Asl et al., 2020; Han et al., 2021) have involved system construction upon a language model pre-trained with a large corpus. Fig. 2 shows the overall architecture of the proposed knowledge-grounded dialogue system. For a given dialogue history, the dialogue state is tracked. The knowledge-seeking turn detection model returns a “true” or “false” prediction, and the domain-tracking model and entity extraction model track and extract the domain and entity, respectively. The extracted entity and domain are leveraged to change the knowledge-seeking turn result or to create knowledge candidates. From the given knowledge candidates, the knowledge selection model selects the most appropriate knowledge for that turn. Finally, the response generator generates a response based on the dialogue history and the selected external knowledge. Each model for each task is trained on pre-trained language models (Zhuang et al., 2021; Radford et al., 2019; Lewis et al., 2020), as explained in detail below.

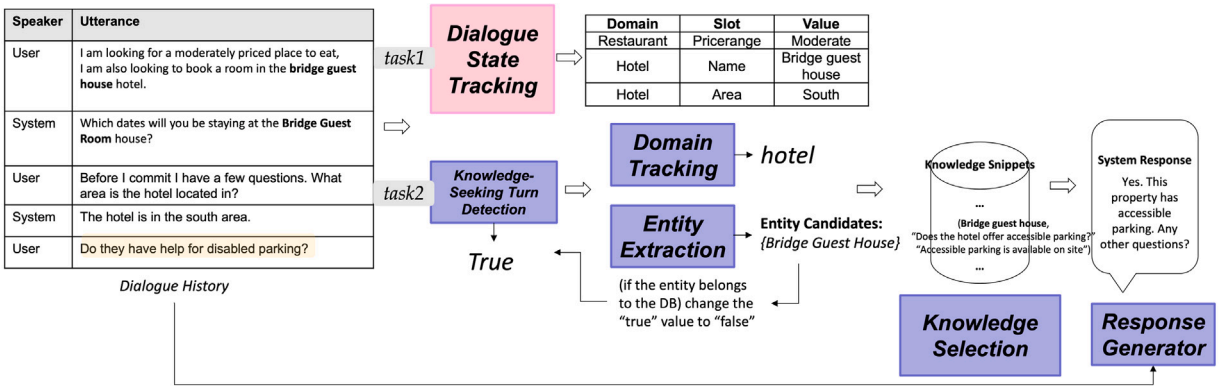


Fig. 2. Overall architecture of proposed knowledge-grounded dialogue system. For the DSTC10 dataset, two tasks are given. (1) As in the pink box of the picture, given the dialogue context, the dialogue state has to be tracked. (2) For the second task, the knowledge-grounded response has to be generated. A sample knowledge-seeking turn is shown for the knowledge entity with the "Bridge Guest House" name, which belongs to the "Hotel" domain. The knowledge-seeking turn detection model predicts "true". The entity-extraction model's predicted entity narrows down the knowledge candidate to be selected as the ones that belong to the predicted entity. If the entity was not found by the entity-extraction model, the tracked domain is used to narrow the candidates to the ones that belong to the domain. Secondly, knowledge selection is made among the narrowed knowledge candidates. Finally, the response is generated with the selected knowledge.

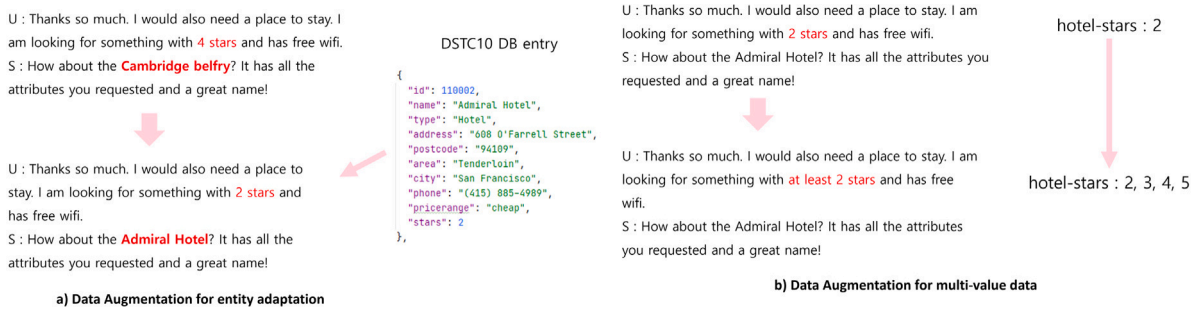


Fig. 3. Example of data augmentation for entity adaptation and multi-value data. (a) The original entity (value), Cambridge Belfry, and its corresponding attributes are replaced with a random DSTC10 entry. (b) By following a multi-value rule, the value of the possible multi-value slot is replaced with a multi-value.

3.1. Dialogue-state tracking with data augmentation

The key to dialogue-state tracking is effective learning of the different domains, slots, and values for each conversation dataset. The MultiWOZ dataset (Eric et al., 2019), in which the dialogue state is automatically annotated for each turn in a task-oriented conversation between real people, is used for dialogue-state tracking model construction. However, dialogue-state tracking models trained with MultiWOZ data may not correctly track dialogue states from other dialogue datasets, having different domains, slots, and values.

Data Augmentation To adapt a high-performance dialogue-state tracking model trained with MultiWOZ data to the domains, slots, and values of the DSTC10 dialogue dataset (Kim et al., 2021), we propose the following data augmentation method:

- Data are augmented by replacing MultiWOZ data entities (values) with entities appearing in the DSTC10 database (DB), as shown in Fig. 3(a). The corresponding attributes are also replaced.

- Fig. 3(b) shows possible multi-value slots in the DSTC10 dataset. As in the above method, for a given dialogue history with possible multi-value slots, human annotators replace the original values with the multi-values.

Revised Slot Types The TripPy model (Heck et al., 2020) has been used in many recent studies (Dai et al., 2021; Li et al., 2020; Yu et al., 2021) to enhance the best-performing model on the MultiWOZ dataset. As shown in Fig. 4, TripPy selects the slot type by which the value for each slot will be determined and, in turn, infers a value according to the determined type. The slot type can be one of the following set elements: none, dontcare, span, inform, refer, True, False. In this study, we propose a revised slot-type set: none, dontcare, span, inform, refer, multi-value. The "multi-value" type has been added so that the model can effectively train the proposed augmented data with multi-values. When the model selects the multi-value type, one or more values are assigned to each slot according to a predefined pattern in the DSTC10 development dataset. For example, in the hotel-stars slot, when a text such as 'a star of at least 2' appears in an utterance of the development dataset, the value '2, 3, 4, 5' is assigned as a pattern. Likewise, 'hotel-day', 'hotel-stars', 'restaurant-pricerange', and 'restaurant-time' slots are assigned with multi-values according to the specific

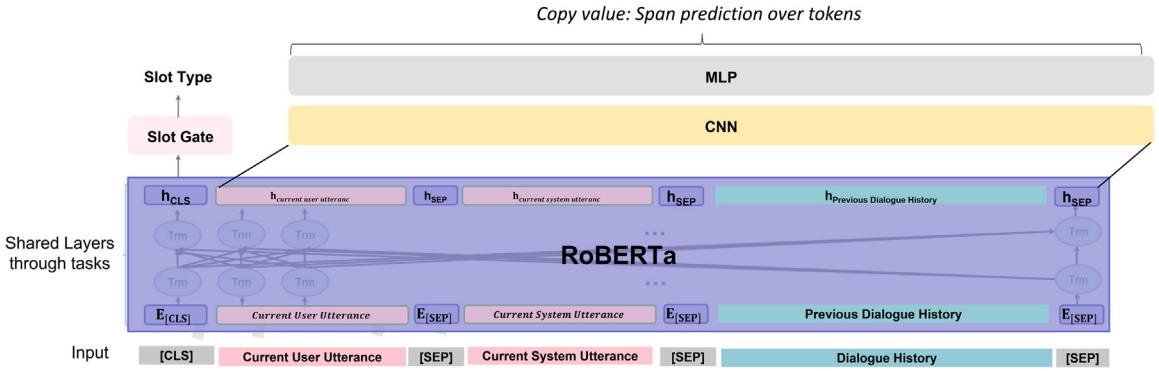


Fig. 4. Revised version of TripPy model. The CNN layer is added to the last layer of the language model and before the multi-layer perceptron (MLP). Here, h, E, [SEP] refer to hidden state of the last layer, an embedding vector, and the separate token, respectively.

knowledge and pattern from the development dataset. For slot types other than multi-value, values are assigned for each slot type in the same way as in the existing TripPy model. That is, for inform type, the starting point and ending point of span are found in the input sequence and the span is assigned as values. For refer type, the value is copied from the dialogue act information, and for the other types (e.g., none, dontcare), its corresponding slot name is regarded as the value. Moreover, the "true" and "false" slot types have been removed because the dialogue states in the DSTC10 dataset do not require slots with these values. In addition to the original TripPy model, we newly stack a convolutional neural network (CNN) (Collobert et al., 2011) layer before the classification layer to represent the input as follows:

$$R_t^{DST} = \text{RoBERTa}([\text{CLS}] \oplus U_t [\text{SEP}] \oplus S_t \oplus [\text{SEP}] \oplus H_t \oplus [\text{SEP}]), \tag{1}$$

$$R_t'^{DST} = \text{CNN}(R_t^{DST}) \tag{2}$$

where R_t^{DST} denotes the hidden state of every token from the RoBERTa model for the dialogue state tracking task given the input at time t. U_t, S_t, and H_t indicate the user utterance at turn t, the system utterance that precedes U_t, and the conversation history, respectively. Our proposed model was named "Multi-TripPy".

3.2. Domain tracking

MultiWOZ-based DSTC10 data include the following possible domain set:train, taxi, hotel, restaurant, attraction. One or more domains can appear within a single conversation. As the last utterance of a conversation does not necessarily provide domain information, the model takes the last five utterances as input. For the domain-tracking task, RoBERTa-large (Zhuang et al., 2021) is used as the pre-trained language model. The classification layer takes the final hidden state of the [CLS] token (Devlin et al., 2019), which is the first position of the input, to the linear layer and then projects a score for all five domains. The model loss can be calculated using the cross-entropy loss, as follows:

$$R_t^{DT} = \text{RoBERTa}([\text{CLS}] \oplus S_{t-2} \oplus U_{t-1} \oplus S_{t-1} \oplus U_t \oplus S_t \oplus [\text{SEP}]), \tag{3}$$

$$g_{DT}(x) = softmax(R_{t,0}^{DT} W_{DT}^{\mathrm{T}}), \tag{4}$$

$$\mathcal{L}_{DT} = -\sum_{x \in D_{DT}} y \log(g_{DT}(x)) \tag{5}$$

where $R_{t,0}^{DT} \in R^H$ is the last layer representation of RoBERTa at the first token (CLS token) with hidden size H, $W_{DT} \in R^{5 \times H}$ is a learnable parameter, and $y \in R^5$ is the domain class indicator. $\mathcal{L}_{DT}$, D_{DT} denote the training loss of the domain tracking task and the domain tracking dataset, respectively.

3.3. Entity extraction

The dialogue text of the DSTC10 dataset may contain typos, as this text is automatic speech recognition output. Moreover, in some cases, the entity name is not stated identically to that in the entity dictionary. Therefore, the entity extraction model is designed to point to the entity start and end, to roughly extract the entity span from the dialogue history. To this end, RoBERTa is trained to maximize the probability of the start and end positions of the label entity in two separate classification layers, as follows:

$$R_t^{EE} = \text{RoBERTa}([\text{CLS}] \oplus TQ \oplus H_t \oplus [\text{SEP}]), \tag{6}$$

$$g_{start}(x) = softmax(R_t^{EE} \mathrm{W}_{start}^{\mathrm{T}}), \tag{7}$$

$$g_{end}(x) = softmax(R_t^{EE} \mathrm{W}_{end}^{\mathrm{T}}), \tag{8}$$

$$\mathcal{L}_{EE} = -\sum_{x \in D_{EE}} y_s \log(g_{start}(x)) + y_e \log(g_{end}(x)) \tag{9}$$

where R_t^{EE} denotes the hidden state of every token from the RoBERTa for the entity extraction model given the input at time t, W_{start}, $W_{end} \in R^{2\times H}$ are learnable parameters used to project the scores for the entity start and end positions, respectively; and y_s, $y_e \in R^L$ are the class indicators for the start and end positions, respectively, along length L of all tokens. TQ is the template question ("What [DOMAIN] are they talking about") that is fed into the domain information input of the domain-tracking model. $\mathcal{L}_{EE}$, D_{EE} denote the training loss of the entity extraction task and the entity extraction dataset, respectively.

Alias Matching To handle cases involving aliases of entity names, we created a dictionary of possible alias names for each entity. In this design, if the extracted span does not yield a precise match for a name in the dictionary of external knowledge entities, and if the alias of an entity in the alias dictionary is found, that entity is added to the entity candidates. We hired three students to annotate aliases of entities for eight weeks and each of the students annotated 300 entity aliases for each week, in total of 7200.

Levenshtein Distance Matching The Levenshtein distance (Levenshtein, 1966) between two words can be defined as the minimum number of single-character edits (insertions, deletions, or substitutions) required to change one word to the other. When the extracted span does not yield an entity that exists in the external knowledge entity dictionary or in the alias dictionary, the Levenshtein distance is calculated for all entity names in the dictionary. Then, the word with the highest score exceeding the threshold of 0.8 is added to the entity candidates. Here, this technique can deal with automatic speech recognition (ASR) error of an entity in the utterance because the Levenshtein distance matching does not require the entity to be exactly matched.

TRIE dictionary-based entity extraction Separately from the entity extraction model introduced above, we created the TRIE (Fredkin, 1960) dictionary to add an entity name as an entity candidate when the first few parts of the actual entity name are found in a character unit. This TRIE-dictionary-based entity extraction method is only used when no entity result has been extracted from the span-based model.

3.4. Knowledge-seeking turn detection

The knowledge-seeking turn detection task determines whether the last user utterance should cause the system to respond with reference to external knowledge. For training, the RoBERTa-large (Zhuang et al., 2021) model takes the last utterance as input. The classification layer takes the hidden state of the CLS token and projects a score from among two values, "true" or "false", to show whether the utterance of the current turn indicates a knowledge-seeking turn. The process is as follows:

$$R_t^{KSTD} = \text{RoBERTa}([\text{CLS}] \oplus U_t \oplus [\text{SEP}]), \tag{10}$$

$$g_{KSTD}(x) = softmax(R_{t,0}^{KSTD} W_{KSTD}^{\mathrm{T}}), \tag{11}$$

$$\mathcal{L}_{KSTD} = -\sum_{x \in D_{KSTD}} y \log(g_{KSTD}(x)) \tag{12}$$

R_t^{KSTD} denotes the hidden state of every token from the RoBERTa for the knowledge-seeking turn detection task given the input at time t, $W_{KSTD} \in R^{2\times H}$ is a learnable parameter and $y \in R^2$ is the knowledge-seeking turn detection class indicator. Note that, in the inference time, the value determined using the above model (either "true" or "false") can be changed according to the following scenarios:

Entity Extraction Result Belonging to DB List If the entity span found in the current conversation history by the entity extraction model does not exist in the entity list of external knowledge, but belongs to the entity list of the general DB, we change the knowledge-seeking turn detection result to "false". The entity span is compared to the entities in the general DB based on the Levenshtein score to determine whether the span matches any entity with a score exceeding the threshold, which is 0.9.

Absence of Entity Extraction Candidates Even if the entity extraction model predicts that there exist no entity candidates in the dialogue history after the knowledge-seeking turn detection model has indicated a knowledge-seeking turn, the knowledge-seeking turn detection model can retain the predicted "true" value. In that case, all entities belonging to the domain identified by the domain-tracking model can be identified as entity candidates; hence, the predicted "true" value will remain valid and the selected entity candidates will be conveyed to the knowledge selection process.

3.5. Knowledge selection

Fig. 5 shows the structure of our proposed knowledge selection model. The DSTC10 knowledge snippets consist of question and answer text pairs. As the last utterance takes the form of a question in a knowledge-seeking turn, the model calculates the similarity between the last utterance and the knowledge-snippet question. The classification layer receives the hidden state of the CLS token from the input representation and projects two similarity scores: the correct and incorrect knowledge-snippet scores. In the training phase, the cross-entropy loss is back-propagated. For the negative examples, the questions of other knowledge snippets belonging to the same entity are used. Four negative examples are learned for each positive example. Here,

$$R_t^{KS} = \text{RoBERTa}([\text{CLS}] \oplus U_t \oplus [\text{SEP}] \oplus \text{KSQ} \oplus [\text{SEP}]), \tag{13}$$

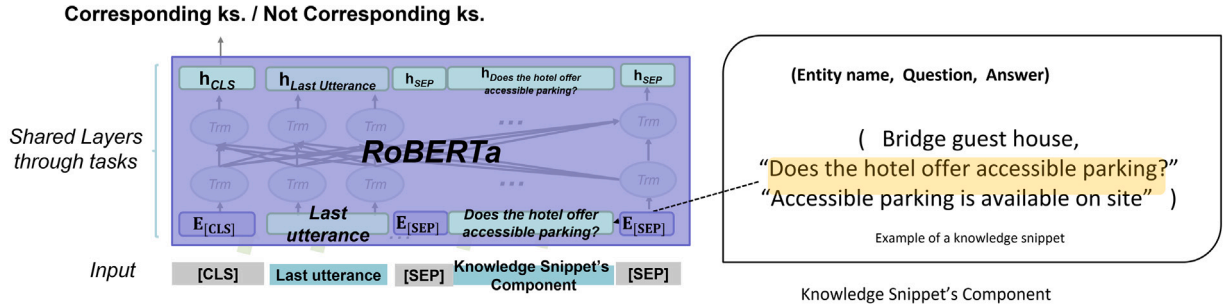


Fig. 5. Model architecture for knowledge selection and knowledge-snippet component. In the example, the last user utterance and the question of the knowledge snippet are taken as input.

$$g_{KS}(x) = softmax(R_{t,0}^{KS} W_{KS}^{\mathrm{T}}), \tag{14}$$

$$\mathcal{L}_{KS} = -\sum_{x \in D_{KS}} y \log(g_{KS}(x)) \tag{15}$$

where R_t^{KS} denotes the hidden state of every token from the RoBERTa for the knowledge selection task given the input at time t, KSQ is the knowledge-snippet question text, $W_{KS} \in R^{2 \times H}$ is a learnable parameter, and $y \in R^2$ is the knowledge-selection class indicator.

In addition to the above setting, we also propose different inputs for models to be trained, where Segments A and B of the inputs are defined as follows: last utterance, knowledge-snippet answer, dialogue history, knowledge-snippet question, dialogue history, knowledge-snippet answer. Each of the models with the four different types of input are trained with the RoBERTa-base (Zhuang et al., 2021) pre-trained language model. At the inference time, for the given knowledge-snippet candidates, the top-scoring knowledge snippet is selected to be passed to the response generator.

Spacy Sentence Similarity At the inference time, if the model predicts a knowledge-seeking turn but no entity candidate exists, the knowledge selection process is given the knowledge snippets of all entities belonging to the domain tracked by the domain-tracking model. As use of a large language model for all these knowledge snippets would be excessively time-consuming, we pre-encode all knowledge snippets using the Spacy[1] sentence encoder before the knowledge selection phase. Then, to select knowledge candidates for scoring by the original knowledge selection model, we calculate the similarity between the pre-encoded knowledge and the last utterance embedding encoded with the Spacy sentence encoder. Finally, the ten highest-scoring candidates are fed into the knowledge selection model and the appropriate knowledge is selected.

Knowledge Reranker After the knowledge selection model extracts the top-five knowledge items based on the calculated score, the final knowledge selection is performed as a final reranking operation. The Reranker model calculates "true"/"false" values for the last utterance and knowledge-snippet question given as inputs to the RoBERTa-Large model. Only the knowledge snippets judged to be true by the reranker, for the existing order of the calculated top-5 knowledge candidates, are finally included among the candidates, and the top-scoring candidate is selected as the final knowledge snippet.

Ensemble For our final model, we use an ensemble of five different models. A sentence-transformer library (Reimers and Gurevych, 2019) provides a BERT-based (Devlin et al., 2019) bi-encoder sentence similarity model for easy fine-tuning. We fine-tune the model using the DSTC10 data by providing input in last utterance, knowledge-snippet question form, to be separately encoded at each encoder. With our RoBERTa-based models, which take the Segment-A and -B inputs in last utterance, knowledge-snippet question, last utterance, knowledge-snippet answer, dialogue history, knowledge-snippet question, and dialogue history, knowledge-snippet answer form, a total of five models are ensembled. All knowledge snippets associated with each of the entity candidates are given as comparison candidates. Each model receives the same knowledge-snippet candidates and outputs the top-five knowledge snippets in order, based on the correct score awarded to each knowledge snippet. Finally, the knowledge-snippet with the highest score based on all models is selected as the output knowledge snippet.

3.6. Response generation

To train a model that generates a response to a given dialogue history using selected knowledge, we finetune GPT-2 (Radford et al., 2019) and BART (Lewis et al., 2020) using the DSTC10 dataset. GPT-2 is comprised of a decoder only; thus, this model takes the concatenated dialogue history and the response. BART inputs the dialogue history to the encoder component and outputs the response in the decoder component. Both models are trained by back-propagating the negative log likelihood loss at the decoder component.

[1] spaCy is an open-source library for natural language processing. https://spacy.io

Table 1
Data statistics for DSTC9 and DSTC10 datasets.

Data	# of Examples		
	Training	Development	Development
DSTC9	71,348	9663	4181
DSTC10	–	263	4191

Table 2
[a]The abbreviated subscripts are defined as follows: dialogue-state tracking (DST), entity extraction (EE), domain tracking (DT), knowledge-seeking turn detection (KSTD), knowledge selection (KS).

Data	# of training examples
D_{DST^a}	21,319
D_{EE}	13,469
D_{DT}	1643
D_{KSTD}	30,724
D_{KS}	255,466

4. Experiments

4.1. Datasets

We conducted experiments on the DSTC10 dataset (Kim et al., 2021), which is a speech-based task-oriented conversation benchmark collected by human interlocutors. In general, when a task-oriented dialogue system performs a specific purpose through conversation with a user, it creates a response by referring to DB data through API calls. In Track 1 of the DSTC9 (Kim et al., 2020) challenge, the aim is to build a task-oriented dialogue system that generates responses, not only by referring to conventional DB information, but also by referring to unstructured external knowledge. To this end, 71 348, 9663, and 4181 training, development, and test examples were released, respectively, as detailed in Table 1. The dataset contained a total of 12 039 knowledge snippets, with each knowledge snippet belonging to one of the domains in the set train, taxi, restaurant, hotel, attraction, and existing in the form of FAQs related to entities belonging to the domain.

In DSTC10, the Track-2 (Kim et al., 2021) challenge had the same theme as the previous DSTC9 Track-1 challenge. In addition to the data of the previous competition, 2292 and 8501 development and test examples, which were converted to text using the Auto Speech Recognition (ASR) model, were additionally released. Moreover, DSTC10 provided development and test sets with 263 and 4191 sample data elements, respectively, for dialogue-state tracking, which is the core feature of a task-oriented dialogue system.

Table 2 summarizes the datasets for the dialogue-state tracking, entity extraction, domain-tracking, knowledge-seeking turn detection, and knowledge selection models. For data augmentation of the dialogue-state tracking task, a total of three human-annotators collected 1397 MultiWOZ-based conversations and a total of 9947 conversations that were changed to DSTC10 entities by referring to the DSTC10 DB. In addition, a total of 2934 multi-value conversations were collected. MultiWOZ 2.1 was used as the training dataset for the TripPy (Heck et al., 2020) model, with a total of 21,319 data samples being employed. All tasks except the dialogue-state tracking task were trained with DSTC9 training and development data, excluding those elements of the DSTC9 test data that overlapped with the DSTC10 development set.

Baselines

DSTC9 Track-1 Baseline (DSTC9) (Kim et al., 2020): This model is the baseline system released by the DSTC9 Track-1 organizers. For each task, i.e., knowledge-seeking turn detection, knowledge selection, and knowledge-grounded response generation, the GPT-2 generation-based model was fine-tuned and used.

DSTC9 Track-1 Winner (Knover) (He et al., 2021): Knover is the model that took first place overall in the DSTC9 Track-1 challenge. Like the DSTC9 model, it uses a generation-based model, but PLATO (Bao et al., 2020) rather than GPT-2. In this study, for the knowledge-selection task, a model was trained using multi-scale negative sampling and a response was generated through beam search decoding.

Evaluation Metrics To evaluate the dialogue-state tracking performance, the joint goal accuracy, which evaluates whether the correct value is predicted for all slots, and the slot accuracy, which evaluates the accuracy of each slot, were considered. Values other than "none" were also evaluated based on the precision, recall, and F1 scores, and values of "none" were evaluated in the same manner. For knowledge-seeking turn detection, the model prediction of whether a given conversation history required external knowledge was evaluated based on the precision, recall, and F1 scores. The knowledge selection performance was evaluated using Recall@k, which evaluated whether the top-k knowledge candidates selected by the model included a correct answer, and MRR@k, which assigned a score by taking the reciprocal of the number of ranks of the predicted correct knowledge among k correct answers. Finally, the response generation task was evaluated based on the BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), and METEOR (Banerjee and Lavie, 2005) scores.

Table 3
Dialogue-state tracking performance on DSTC10 test dataset.

Dialogue-state Tracking setting	Joint goal accuracy	Slot accuracy
TripPy	0.39%	70.52%
Multi-TripPy	4.11%	75.41%
+ data augmentation	8.24%	80.58%
+ post-processing	**15.8%**	**85.59%**

Table 4
Domain tracking performance on DSTC10 development dataset.

Domain tracking Setting	DSTC10 DEV Accuracy
(1) RoBERTa-base	88.46%
(2) RoBERTa-large	91.35%
(2) + MultiWOZ data augmentation	88.46%
(2) + data pre-process	**94.23%**

Table 5
The overall performance on DSTC10 test dataset. R@1, R@5: Recall@1 and Recall@5, respectively; MET.: METEOR; RG-1 and RG-L: ROUGE-1 and ROUGE-L, respectively.

	Turn detection			Knowledge selection			Generation				
	Prec[a]	Recall	F1	MRR@5	R@1	R@5	BLEU1	BLEU4	MET	RG-1	RG-L
Knover	89.67%	67.35%	76.92%	55.74%	49.50%	64.72%	12.48%	1.53%	13.64%	15.19%	12.29%
DSTC9	90.17%	71.16%	79.54%	52.30%	45.83%	62.52%	11.53%	0.75%	12.15%	14.43%	11.43%
Proposed	**92.14%**	**92.68%**	**92.41%**	**64.64%**	**62.04%**	**68.47%**	**37.98%**	**13.73%**	**45.14%**	**44.2%**	**41.54%**

[a]Prec.: Precision.

4.2. Dialogue-state tracking

Table 3 details the performance of the proposed dialogue-state tracking model. As DSTC 10 data have different slots and values to the MultiWOZ dataset, the TripPy model trained with MultiWOZ data had a joint goal accuracy close to zero. Note that the original TripPy cannot handle the multi-value included in the DSTC10 dataset. Whenever the data contains the multi-value instances, the joint-goal accuracy has to be zero even if the model predicted every other values correctly. Moreover, different slots are added in the DSTC10 dataset, thus affects the joint goal accuracy. Our proposed model, Multi-TripPy, which reflected local information more effectively than TripPy through addition of a CNN (Collobert et al., 2011) layer, was trained by augmentation with multi-value data. Multi-TripPy achieved a ten-fold improvement in the joint goal accuracy metric compared to TripPy. Furthermore, if data augmentation was implemented for the training data, to change the MultiWOZ entities to DSTC10 entities, the performance improvement was doubled compared to Multi-TripPy. After the model's prediction is made, the followings are done as post-processing: (1) Pre-defined pattern for fixing the entity name is checked (e.g., a predicted value 'adc' is replaced with 'adc theatre' referring to our pattern map). (2) If the predicted value of name slot exists in the database file, we assume the model's prediction of the value is correct because if the model's prediction has no ASR error, then it is highly likely that the predicted entity name is correct. Thus, using the predicted name value, we refer to the database and copy other attributes, such as 'area' and 'type' slots' values to the model's prediction. For instance, if the model predicted the one of the name slots as 'Inn at Union Square, a Greystone Hotel' and this value exists in the database, we look for the other attributes. In the database, the 'area' slot of this hotel is 'Union Square' and our model's predicted value is replaced with the correct value when they are different. (3) When the model predicts the 'book' slot as true, we assume there must exist a place for booking. However, in cases where the value of the 'name' slot does not exist, we look for a name entity using our pre-defined TRIE dictionary. For example, when the model's prediction for 'hotel-book' slot as 'yes' but the name of the hotel for booking is absent, we search for a matching hotel name using the TRIE dictionary. If the dialogue history contains 'Inn at Union Square, a Greystone Hotel', then it is extracted by the search process and assigned to the 'hotel-name' slot. Through post-processing, a joint goal accuracy of 15.8%p was achieved.

4.3. Knowledge-grounded dialogue tasks

Overall Score Table 5 reports the overall performance of each model on the DSTC10 test dataset. Our proposed model outperformed the previous state-of-the-art model (He et al., 2021) of DSTC9 (i.e., Knover) in terms of the precision, recall, and F1 scores for the knowledge-seeking turn detection task. In particular, a large difference was apparent for the recall score, at 21.52%p. The number of knowledge-seeking turns (23,734) in the training set was approximately three times more than those turns that did not require external knowledge (61,195). If a model naively followed this data distribution, it could naturally obtain a high precision score by lowering the model tendency to return a "true" prediction. However, our model achieved a higher precision score than the other

Table 6
Entity extraction model performance on DSTC10 development dataset.

Entity extraction method	DSTC10 DEV	
	Recall@5	IPP score
TRIE	77.88%	46.71%
RoBERTa-large	91.35%	91.35%
TRIE + RoBERTa-large	**93.27%**	**92.31%**

Table 7
Knowledge-seeking turn detection model performance on DSTC10 test dataset. [a]Models (4) and (5) added post-processing for two cases: if the entity found by the entity extraction model belonged to the DB, and if no entity was found. T → F indicates that the original true value of the knowledge-seeking turn detection model was replaced with the false value.

Knowledge-seeking Turn Detection Method	DSTC10 TEST		
	Precision	Recall	F1
(1) Knover	89.67%	67.35%	76.92%
(2) DSTC9	90.17%	71.16%	79.54%
(3) RoBERTa-large	85.92%	**94.73%**	90.11%
(4) RoBERTa-large + T → F (Entity in DB) T → F (Entity not found)[a]	**95.21%**	81.55%	87.85%
(5) RoBERTa-large + T → F (Entity in DB) Spacy (Entity not found)	92.14%	92.68%	**92.41%**

models and, simultaneously, a significantly higher recall score, with a performance difference of 12.87%p in the final F1 score. The knowledge selection model exhibited the highest performance in terms of all evaluation metrics, i.e., MRR@5, Recall@1, and Recall@5. Our proposed system used GPT-2 (Radford et al., 2019) as the generation model, and substantially outperformed the other models in terms of the overall generation score.

Analysis

4.3.1. Domain tracking

The domain-tracking performance was investigated using the settings given in Table 4. The RoBERTa-large model exhibited a 2.89%p accuracy improvement compared to the RoBERTa-base model. With the original MultiWOZ data, we automatically annotated the domain information for every turn with a newly appearing dialogue-state domain. In the table, the result is reported as "(2) + MultiWOZ data augmentation". Although the volume of training data increased and data augmentation was performed, the 2.89%p accuracy enhancement disappeared. However, an additional accuracy improvement of 2.88%p (i.e., 5.77%p overall) was achieved for learning with pre-processed data. As pre-processing, questions and answers that could interfere with interpretation of the conversation domain were removed.

4.3.2. Entity extraction

As various entities can appear multiple times in a single conversation, identification of exactly one entity for which external knowledge must be referenced is challenging. When only the TRIE (Fredkin, 1960) dictionary was used to extract the entity from the dialogue history, the Recall@5 score was 77.88%p, as reported in Table 6. Using the TRIE dictionary, the possible entities were safely included in the entity candidates; however, this number had the potential to grow excessively. To evaluate the accuracy with which an entity is included in the entity candidates per the number of predictions, we propose an "included per predictions" (IPP) score. If the correct entity is included in the entity candidates, the IPP score is obtained by dividing 1 by the number of inferences; otherwise, this score is 0. Here, the span-based model using RoBERTa-large had a very high Recall@5 score and an IPP score that was 44.64%p higher than that of the TRIE model. Using both the TRIE dictionary and the span-based model, additional improvements of 1.92%p and 0.96%p in the Recall@5 and IPP scores, respectively, were obtained.

4.3.3. Knowledge-seeking turn detection

Table 7 reveals how leveraging the entity and domain information in the knowledge-seeking turn detection model can improve performance. The model fine-tuned with DSTC10 data labelled with the knowledge-seeking turn on the RoBERTa-large language model outperformed the two baselines (Kim et al., 2020; He et al., 2021) and exhibited the highest recall score. However, the precision score of this model was 6.22%p lower than that of Model (5) in the table. Note that, in Models (4) and (5), a knowledge-seeking turn prediction of "true" was changed to "false" when the entity found by the entity extraction model did not exist in the knowledge-snippet entity list but did exist in the general DB. This method of post-processing improved the precision score because it reduced the number of true predictions. If the entity extraction model found no entity candidates, but the knowledge-seeking turn detection model returned a "true" prediction for the turn, in Model (4), the "true" value became "false". In contrast,

Table 8
Knowledge selection model performance for different model and input types.

Knowledge selection Models	Input type	DSTC10 DEV		
		R@1	R@2	R@5
Sentence-BERT (Reimers and Gurevych, 2019)	Lu2q	75.96%	82.69%	89.42%
RoBERTa-base	Lu2q	**83.65%**	90.38%	**93.27%**
	Lu2a	82.69%	87.50%	91.35%
	H2q	82.69%	**91.35%**	91.35%
	H2a	79.81%	86.54%	92.31%

Table 9
Knowledge selection performance on DSTC10 test dataset.

Knowledge selection models	Input type	DSTC10 TEST		
		R@1	R@2	R@5
Knover	–	55.74%	49.50%	64.72%
DSTC9	–	52.30%	45.83%	62.52%
RoBERTa-base	Lu2q	60.21%	57.37%	65.11%
	H2q	60.86%	58.1%	65.69%
+ Reranking	Lu2q	61.55%	59.27%	65.11%
+ Ensemble	All	**64.64%**	**62.04%**	**68.47%**

Table 10
Response generation performance on DSTC10 test dataset.

Response generation model	DSTC10 TEST			
	BLEU1	BLEU4	METEOR	ROUGE1
Knover	12.48%	1.53%	13.64%	14.19%
DSTC9	11.53%	0.75%	12.15%	14.43%
(Proposed) BART	27.05%	2.98%	31.67%	33.24%
(Proposed) DialoGPT	26.91%	4.95%	30.72%	34.00%
(Proposed) GPT-2	**37.98%**	**13.73%**	**45.14%**	**44.2%**

for Model (5), every knowledge snippet that belonged to the domain predicted by the domain-tracking model was added to the knowledge candidates. Then, each embedding of the knowledge candidates, which were pre-encoded by the Spacy sentence encoder, was compared with the encoded dialogue history. The ten highest-scoring candidates were passed to the next knowledge selection stage. Both Models (4) and (5) achieved high precision scores, with Model (4) having the highest overall, at 95.21%p. However, the recall performance decreased by 11.18%p compared to that of the RoBERTa-large model. Model (5) achieved the best F1 score of 92.41%p by leveraging the domain prediction value of the domain-tracking model to select the knowledge candidates in a true knowledge-seeking turn.

4.3.4. Knowledge selection

Knowledge selection scores obtained on the DSTC10 development set for different models and different input types are reported in Table 8. Here, "Lu", "H", "q", and "a" represent the last utterance of the dialogue, dialogue history, knowledge-seeking question, and knowledge-based answer, respectively. As apparent from the table, the case employing the RoBERTa-base model achieved higher overall performance than a model fine-tuned with Sentence-BERT (Reimers and Gurevych, 2019). The maximum performance difference was 7.69%p for Recall@1. As regards the RoBERTa-base training, the RoBERTa model with a knowledge-snippet question as Segment-B input exhibited higher performance than the model with a knowledge-snippet answer as Segment-B input. The same results were obtained when the Segment-A input corresponded to the dialogue history or last utterance. We presume this is because the last utterance had similar form and meaning to the knowledge-snippet question. In addition, the model taking the last utterance as the Segment-A input and the knowledge-snippet question as Segment-B input achieved a higher Recall@1 score than the model with the full dialogue history as the Segment-A input.

Table 9 reports the knowledge-selection results on the DSTC10 test dataset. For the models with the last utterance and knowledge-snippet question as the Segment-A and -B inputs, respectively, the performance exceeded that of the baseline models. In particular, the Recall@1 score was 8.6%p higher than that of Knover (He et al., 2021). When the reranking model was added to the Lu2q model, an additional 1.17%p improvement in Recall@1 was obtained. Through collective application of the five proposed models and reranking, an additional improvement in Recall@1 of 2.77%p was obtained.

4.3.5. Response generation

For the response generation model, we investigated the performance of both the BART (Lewis et al., 2020), DialoGPT (Zhang et al., 2020), and GPT-2 (Radford et al., 2019) models, and compared them to the two baseline models, which were fine-tuned on the

DSTC9 dataset with the GPT-2 model as a generation model. However, the generation scores of the baseline models were relatively low because of their lower performance in the tasks implemented before the generation stage. When all tasks in the previous step were performed using the same model, and the BART, DialoGPT, and GPT-2 models were compared for the generation task only, the GPT-2 model yielded significantly improved performance in terms of all BLEU, METEOR, and ROUGE generation evaluation metrics (see Table 10).

4.3.6. Computational cost

Our work consists of various types and numbers of models which can be concerning for a real-world dialogue system. The DSTC10 dataset is released for the competition purpose, and it is common to ensemble the models to boost the performance. We have found that this ensemble process for our system is the most time-consuming during the inference. In the knowledge selection process where the ensemble is used, it spends 67% of the overall elapsed time for the inference. However, when the ensemble is removed, it significantly drops down to 26.52% of the overall inference time. Removing the ensemble, on the other hand, does not severely affect the performance. The MRR@5, recall@1, and recall@5 score only drop by 2.39%, 3.28%, and 1.7% of the ensemble model. Therefore, if our system focuses more on the real-world usage of the dialogue rather than the competition environment, the system can be practically viable both in computational and qualitative respect. This is the result of using only 2 RTX Titan 3090 GPUs and calculating with only 8 batch sizes in total due to our limited resources. Thus, we expect the overall process can be much shortened with more powerful resources.

5. Conclusion

A knowledge-grounded dialogue system should effectively identify when and which external knowledge is required for the current turn, and should effectively generate responses based on the selected external knowledge. In this study, performance improvements were achieved for each task performed by a knowledge-grounded dialogue system by exploiting the fact that the external knowledge associated with a question–answer pair is closely related to a specific domain and entity. In addition, we proposed a high-performance dialogue-state tracking model adapted to the specific slots and values occurring in external knowledge-based dialogue datasets. This study demonstrated the effectiveness of leveraging the domain and entity information when a knowledge-seeking turn is incorrectly identified and, also, how this information can be used to narrow the scope of the knowledge to be searched.

In future research, correction of a false-negative prediction for a knowledge-seeking turn using knowledge-based entity and domain information may be explored. In addition, as knowledge is provided in various forms (e.g., in documents, knowledge graphs, and question–answer pairs), effective use of external knowledge in a general form is an important challenge for future work.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The authors do not have permission to share data.

Acknowledgements

This work was supported in part by Institute of Information & Communications Technology Planning & Evaluation (IITP), South Korea grant funded by the Korea Government (MSIT) (No. 2020-0-00368, A Neural-Symbolic Model for Knowledge Acquisition and Inference Techniques), and in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF-2020R1A2C2100362).

References

Banerjee, S., Lavie, A., 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In: Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/Or Summarization. Association for Computational Linguistics, Ann Arbor, Michigan, pp. 65–72.

Bao, S., He, H., Wang, F., Wu, H., Wang, H., 2020. PLATO: Pre-trained dialogue generation model with discrete latent variable. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, Online, pp. 85–96. http://dx.doi.org/10.18653/v1/2020.acl-main.9.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P., 2011. Natural language processing (almost) from scratch. J. Mach. Learn. Res. 12 (null), 2493–2537.

Dai, Y., Li, H., Li, Y., Sun, J., Huang, F., Si, L., Zhu, X., 2021. Preview, attend and review: Schema-aware curriculum learning for multi-domain dialogue state tracking. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers). Association for Computational Linguistics, Online, pp. 879–885. http://dx.doi.org/10.18653/v1/2021.acl-short.111.

Devlin, J., Chang, M.-W., 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). Association for Computational Linguistics, Minneapolis, Minnesota, pp. 4171–4186. http://dx.doi.org/10.18653/v1/N19-1423.

Dinan, E., Roller, S., Shuster, K., Fan, A., Auli, M., Weston, J., 2019. Wizard of Wikipedia: Knowledge-powered Conversational Agents. In: Proceedings of the International Conference on Learning Representations (ICLR).

Eric, M., Goel, R., Paul, S., Sethi, A., Agarwal, S., Gao, S., Hakkani-Tur, D., 2019. MultiWOZ 2.1: Multi-domain dialogue state corrections and state tracking baselines. arXiv preprint arXiv:1907.01669.

Fredkin, E., 1960. Trie memory. Commun. ACM 3 (9), 490–499. http://dx.doi.org/10.1145/367390.367400.

Galley, M., Brockett, C., Gao, X., Dolan, B., Gao, J., 2019. End-to-end conversation modeling: DSTC7 task 2 description. In: AAAI Workshop Dialog System Technology Challenge (DSTC).

Ghazvininejad, M., Brockett, C., Chang, M.-W., Dolan, B., Gao, J., Yih, W.-t., Galley, M., 2018. A knowledge-grounded neural conversation model. In: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 32.

Guu, K., Lee, K., Tung, Z., Pasupat, P., Chang, M.-W., 2020. REALM: Retrieval-augmented language model pre-training. arXiv:2002.08909.

Han, J., Hong, T., Kim, B., Ko, Y., Seo, J., 2021. Fine-grained post-training for improving retrieval-based dialogue systems. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics, Online, pp. 1549–1558.

He, H., Lu, H., Bao, S., Wang, F., Wu, H., Niu, Z., Wang, H., 2021. Learning to select external knowledge with multi-scale negative sampling. arXiv:2102.02096.

Heck, M., van Niekerk, C., Lubis, N., Geishauser, C., Lin, H.-C., Moresi, M., Gasic, M., 2020. Trippy: A triple copy strategy for value independent neural dialog state tracking. In: Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue. Association for Computational Linguistics, 1st virtual meeting, pp. 35–44.

Hosseini-Asl, E., McCann, B., Wu, C.-S., Yavuz, S., Socher, R., 2020. A simple language model for task-oriented dialogue. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (Eds.), Advances in Neural Information Processing Systems, Vol. 33. Curran Associates, Inc., pp. 20179–20191.

Huang, H.-Y., Choi, E., tau Yih, W., 2019. FlowQA: Grasping flow in history for conversational machine comprehension. In: International Conference on Learning Representations.

Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., Yih, W.-t., 2020. Dense passage retrieval for open-domain question answering. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics, Online, pp. 6769–6781. http://dx.doi.org/10.18653/v1/2020.emnlp-main.550.

Kim, S., Eric, M., Gopalakrishnan, K., Hedayatnia, B., Liu, Y., Hakkani-Tur, D., 2020. Beyond domain APIs: Task-oriented conversational modeling with unstructured knowledge access. arXiv preprint arXiv:2006.03533.

Kim, S., Liu, Y., Jin, D., Papangelis, A., Gopalakrishnan, K., Hedayatnia, B., Hakkani-Tur, D., 2021. "How robust r u?": Evaluating task-oriented dialogue systems on spoken conversations. arXiv:2109.13489.

Levenshtein, V.I., 1966. Binary codes capable of correcting deletions, insertions and reversals.. Sov. Phys. Doklady 10 (8), 707–710, Doklady Akademii Nauk SSSR, V163 No4 845-848 1965.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L., 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, Online, pp. 7871–7880. http://dx.doi.org/10.18653/v1/2020.acl-main.703.

Li, S., Yavuz, S., Hashimoto, K., Li, J., Niu, T., Rajani, N., Yan, X., Zhou, Y., Xiong, C., 2020. CoCo: Controllable counterfactuals for evaluating dialogue state trackers. ArXiv arXiv:2010.12850.

Lin, C.-Y., 2004. ROUGE: A package for automatic evaluation of summaries. In: Text Summarization Branches Out. Association for Computational Linguistics, Barcelona, Spain, pp. 74–81.

Mehri, S., Eric, M., Hakkani-Tur, D., 2020. DialoGLUE: A natural language understanding benchmark for task-oriented dialogue. ArXiv arXiv:2009.13570.

Moon, S., Shah, P., Kumar, A., Subba, R., 2019. OpenDialKG: Explainable conversational reasoning with attention-based walks over knowledge graphs. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics.

Papineni, K., Roukos, S., Ward, T., Zhu, W.-J., 2002. Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, Philadelphia, Pennsylvania, USA, pp. 311–318. http://dx.doi.org/10.3115/1073083.1073135.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., 2019. Language models are unsupervised multitask learners.

Reddy, S., Chen, D., Manning, C.D., 2019. CoQA: A conversational question answering challenge. Trans. Assoc. Comput. Linguist. 7, 249–266. http://dx.doi.org/10.1162/tacl_a_00266.

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). Association for Computational Linguistics, Hong Kong, China, pp. 3982–3992. http://dx.doi.org/10.18653/v1/D19-1410.

Saeidi, M., Bartolo, M., Lewis, P., Singh, S., Rocktäschel, T., Sheldon, M., Bouchard, G., Riedel, S., 2018. Interpretation of natural language rules in conversational machine reading. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Brussels, Belgium, pp. 2087–2097. http://dx.doi.org/10.18653/v1/D18-1233.

Su, Y., Shu, L., Mansimov, E., Gupta, A., Cai, D., Lai, Y., Zhang, Y., 2021. Multi-task pre-training for plug-and-play task-oriented dialogue system. CoRR abs/2109.14739, arXiv:2109.14739.

Wolf, T., Sanh, V., Chaumond, J., Delangue, C., 2019. Transfertransfo: A transfer learning approach for neural network based conversational agents. arXiv:1901.08149.

Wu, C.-S., Madotto, A., Liu, W., Fung, P., Xiong, C., 2021. Qaconv: Question answering on informative conversations. arXiv preprint arXiv:2105.06912.

Yu, T., Zhang, R., Polozov, O., Meek, C., Awadallah, A.H., 2021. SCoRE: Pre-training for context representation in conversational semantic parsing. In: International Conference on Learning Representations.

Zhang, S., Dinan, E., Urbanek, J., Szlam, A., Kiela, D., 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). Association for Computational Linguistics, Melbourne, Australia, pp. 2204–2213. http://dx.doi.org/10.18653/v1/P18-1205.

Zhang, Y., Sun, S., Galley, M., Chen, Y.-C., Brockett, C., Gao, X., Gao, J., Liu, J., Dolan, B., 2020. DialoGPT: Large-scale generative pre-training for conversational response generation. In: ACL, System Demonstration.

Zhou, K., Prabhumoye, S., Black, A.W., 2018. A dataset for document grounded conversations. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing.

Zhuang, L., Wayne, L., Ya, S., Jun, Z., 2021. A robustly optimized BERT pre-training approach with post-training. In: Proceedings of the 20th Chinese National Conference on Computational Linguistics. Chinese Information Processing Society of China, Huhhot, China, pp. 1218–1227.