Unstructured Knowledge Access in Task-oriented Dialog Modeling using Language Inference, Knowledge Retrieval and Knowledge-Integrative Response Generation

Mudit Chaudhary\(^1\)*, Borislav Dzodzo\(^1\), Sida Huang\(^1\), Chun Hei Lo\(^1\), Mingzhi Lyu\(^1\), Lun Yiu Nie\(^1\), Jinbo Xing\(^1\), Tianhua Zhang\(^2\), Xiaoying Zhang\(^1\), Jingyan Zhou\(^1\), Hong Cheng\(^1,2\), Wai Lam\(^1,2\), Helen Meng\(^1,2\)

\(^1\)The Chinese University of Hong Kong
\(^2\)Centre for Perceptual and Interactive Intelligence
muditchaudhary@cuhk.edu.hk, {bdzodzo, chlo, zhangxy, jyzhou, hcheng, wlam, hmmeng}@se.cuhk.edu.hk
{sdhuang, lynie, thzhang}@link.cuhk.edu.hk, {mzlyu, jbxing}@cse.cuhk.edu.hk

Abstract

Dialog systems enriched with external knowledge can handle user queries that are outside the scope of the supporting databases/APIs. In this paper, we follow the baseline provided in DSTC9 Track 1 and propose three subsystems, KDEAK, KnowleDgEFactor, and Ens-GPT\(^*\), which form the pipeline for a task-oriented dialog system capable of accessing unstructured knowledge. Specifically, KDEAK performs knowledge-seeking turn detection by formulating the problem as natural language inference using knowledge from webpages, databases and FAQs. KnowleDgEFactor accomplishes the knowledge selection task by formulating a factorized knowledge/document retrieval problem with three modules performing domain, entity and knowledge level analyses. Ens-GPT generates a response by first processing multiple knowledge snippets, followed by an ensemble algorithm that decides if the response should be solely derived from the external knowledge or databases/APIs. Therefore, we set out to create a pipeline for a task-oriented dialog system capable of accessing unstructured knowledge. Specifically, KDEAK performs knowledge-seeking turn detection by formulating the problem as natural language inference using knowledge from webpages, databases and FAQs. KnowleDgEFactor accomplishes the knowledge selection task by formulating a factorized knowledge/document retrieval problem with three modules performing domain, entity and knowledge level analyses. Ens-GPT generates a response by first processing multiple knowledge snippets, followed by an ensemble algorithm that decides if the response should be solely derived from the external knowledge or databases/APIs. Therefore, we set out to create a pipeline for a task-oriented dialog system capable of accessing unstructured knowledge. Specifically, KDEAK performs knowledge-seeking turn detection by formulating the problem as natural language inference using knowledge from webpages, databases and FAQs. KnowleDgEFactor accomplishes the knowledge selection task by formulating a factorized knowledge/document retrieval problem with three modules performing domain, entity and knowledge level analyses. Ens-GPT generates a response by first processing multiple knowledge snippets, followed by an ensemble algorithm that decides if the response should be solely derived from the external knowledge or databases/APIs.

Table 1: Example of a domain-wide (line-1) and an entity-specific knowledge snippet (line-2). \(T\), \(B\) represent the title and the body.

| Domain  | Entity | Snippet |
|---------|--------|---------|
| Train   | –      | \(T\): Is there a charge for using WiFi? \(B\): Wifi is available free of charge. |
| Hotel   | Avalon | \(T\): Are pets allowed on site? \(B\): Pets are not allowed at avalon. |

\(^*\)All authors have contributed equally.

Copyright © 2021, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

1 Introduction

By incorporating the external knowledge sources available on webpages, task-oriented dialog systems can be empowered to handle various user requests that are outside the coverage of their APIs or databases. Therefore, we set out to create a dialog system that outperforms the Ninth Dialog System Technology Challenge (DSTC9) Track 1 baseline (Kim et al. 2020; Gunasekara et al. 2020). The baseline method is a pipeline composed of three tasks: the first task recognizes if a dialog response requires knowledge outside of a provided MultiWOZ 2.1 database (Eric et al. 2019). If so, the second task then retrieves the most relevant knowledge snippets from an external knowledge base, which are subsequently used together with the dialog context to generate a response in the third task. Specifically, all the three tasks are handled by the variants of pre-trained GPT2 models (Vaswani et al. 2017; Wolf et al. 2019).

Formally, the external knowledge base \(K\) is composed of knowledge snippets \(k_1, \ldots, k_n\); \(D\) is the set of all domains. For the DSTC9 Track 1 Training Set, \(D = \{\text{hotel, restaurant, train, taxi}\}\). Table 1 shows examples of the two types of knowledge, namely a domain-wide knowledge snippet directly under a specific domain \(d_i = \text{train}\), and an entity-specific knowledge snippet of entity \(e_i = \text{Avalon}\), which belongs to the domain hotel. \(D_w\) and \(D_e\) refer to the domains that contain only domain-wide and only entity-specific knowledge snippets respectively, \(D_w \cup D_e = D\) and \(D_w \cap D_e = \emptyset\). A snippet \(k_i\) consists of a title (question) and a body (answer). A knowledge snippet \(k_i\) is considered in-domain (ID) if its domain \(d_i\) was seen during the training of the models; otherwise, it is considered out-of-domain (OOD). The dialog history \(U_t = \{u_{t-w+1}, \ldots, u_{t-1}, u_t\}\) contains utterances \(u_t\) where \(t\) is the time step of the current user utterance and \(w\) is the size of dialog context window. Responses to this dialog are found in the ground truth \(\hat{r}_{t+1}\) or they can be generated by our system \(\hat{r}_{t+1}\).

We created a transparent, factorized, generalisable and...
knowledge-grounded task-oriented conversational system with code available at http://bit.ly/2I5y3KW. Multiple information retrieval hypotheses are considered when constructing the response and this significantly improves results. When the three tasks are integrated they significantly outperform the baseline in terms of automated metrics.

2 Methodology

2.1 Task 1 – Knowledge-seeking Turn Detection

As mentioned earlier, Task 1 classifies whether information from the database or external knowledge is required to answer a user’s query.

We introduce KDEAK (Knowledge-seeking turn detection using Domain, Entity, API/DB and Knowledge) shown in Figure 1. The domain classifier helps the entity classifier determine the dialog’s relevant entity. We generate candidate information snippets from the selected entity’s database and knowledge. The knowledge classifier ranks and classifies the candidate information snippets to determine whether the database or knowledge answers the user’s query. In the subsequent sections, we illustrate our KDEAK’s modules using the example from Table 2. What differentiates KDEAK from Task 2 is that its Domain Classifier can identify domains in the non-knowledge-seeking turns and the Knowledge Classifier’s ability to select the relevant API/DB information.

NLI Problem Formulation. We formulate Task 1’s domain classification, and knowledge classification problems as a Natural Language Inference (NLI) problem (Dagan, Glickman, and Magnini 2005). The NLI problem deals with a pair of statements – hypothesis and premise. Given the premise, it determines whether the hypothesis is True (i.e., an entailment), False (i.e., a contradiction), or Undetermined. For example, if “I want to book a hotel” is the premise, the hypotheses “The user wants to book a hotel” is True and “The user wants to book a taxi” is False. We leverage a pre-trained NLI model (Lewis et al. 2019) for classification in Task 1. We use the last $N_{dialog}$ turns for premise generation. We pair each premise with a set of generated candidate hypotheses using domain and knowledge labels. We find the NLI approach more robust against unseen domains as compared to the baseline.

Module 1 - Domain Classifier. This module classifies the dialog turn’s relevant domain. We generate the premise using the following premise template – “Assistant says $system_response$. User says $user_response$” in each dialog turn to distinguish between user and system response ($N_{dialog} = 2$). Based on the example in Table 3, the premise will be “Assistant says The SW ... book? User says Yes ... there?” We pair the premise with a candidate hypothesis for each domain $d_i \in D$ using the hypothesis template – “The user is asking about $d_i$.” We feed these pairs into the NLI model to find the most probable domain by performing a softmax on each candidate hypothesis’ output entailment probability. The domain $d_i$ with the maximum entailment probability is selected for the dialog turn.

We use Bidirectional and Auto-Regressive Transformers (BART) model (Lewis et al. 2019) initially fine-tuned on MultiNLI (Williams, Nangia, and Bowman 2018). We further fine-tune our model on MultiWOZ2.2 (Zang et al. 2020) and DSTC9 Track 1 Training Set on all eight domains of MultiWOZ2.2. For training, we generate the premise and hypotheses using the templates mentioned above. Each premise with ground-truth domain $d_i$ is paired with the hypothesis corresponding to $d_i$ and marked as entailment. We also pair the same premise with the remaining $|D| - 1$ hypotheses and mark them as contradiction. For inference, we use Huggingface’s (Wolf et al. 2019) ‘classification-as-NLI’ based zero-shot-classification pipeline.

Module 2 - Entity Classifier. The Entity Classifier uses the selected domain from the Domain Classifier to further process the dialog turn in focus. We devise a Surface Matching Algorithm (SMA) to match the possible entities within the dialog history with carefully designed heuristics, based on the intuition that the later the entity appears in the dialog history, the more likely it is the target. Approximate string matching is also incorporated into the algorithm to enhance its robustness to alias matching and misspelling. For instance, SMA is capable of retrieving the entities A & B Guest House from seeing ‘A and B’, Avalon from seeing ‘Avalon.’ The selected domain label helps reduce the entity search space. Following on with our example (see Table 2), the entities corresponding to the hotel domain are searched to see if they occur in the dialog turn. Consequently, the matching algorithm identifies SW Hotel as the entity.

Module 3 - Candidate Information Generator (CIG). Given the identified entity from the Entity Classifier, this module consolidates the relevant database snippets and knowledge snippets for the entity and places them into an information candidate pool $C_{entity}$, which will be used by the Knowledge Classifier in the subsequent step. As we observe in Table 2, database snippets are not natural sounding like knowledge-snippets, so we pre-process them using suitable formatting templates before adding them to $C_{entity}$. Based on Table 2’s example, the database snippet – {$name: SW Hotel, postcode: 94133$} becomes “Postcode for SW Hotel is 94133.” We also add pseudo-candidates to $C_{entity}$ to deal with cases where information is not present in either database or knowledge, e.g., “Goodbye”, “I want to book a hotel”, “Thanks”, etc. Following on with our example, CIG
Table 2: Excerpt of last 2 dialog turns from hotel domain with relevant knowledge snippet (T: title, B: body) and database entry.

| Speaker | Utterance (ui) | Knowledge Snippet (ki) | Database Entry |
|---------|---------------|------------------------|----------------|
| Assistant | Would you like to book the SW hotel? | - | name: SW Hotel |
| User | Yes, I can reach SW hotel by taxi. What breakfast options are available there? | T: Does SW Hotel offer breakfast? B: No, we don’t offer breakfast. | address: 615 Broadway postcode: 94133 type: Hotel |

Table 3: Evaluation results of Task 1 on knowledge-seeking turn detection on DSTC9 Track 1 Validation and Test Sets. Submitted system using $N_{dialog} = 1$ without premise template. Improved system using $N_{dialog} = 2$ and premise template.

| Method      | Validation Set | Test Set |
|-------------|---------------|----------|
|             | Accuracy | Precision | Recall | F1       | Accuracy | Precision | Recall | F1       |
| GPT2-Baseline | 0.995    | 0.999     | 0.982  | 0.991    | 0.946    | 0.993     | 0.892  | 0.940    |
| KDEAK$^*$    | 0.993    | 0.980     | $^*$   | 0.993    | 0.924    | 0.989     | 0.849  | 0.914    |
| KDEAK$^*$    | 0.994    | 0.993     | 0.986  | 0.989    | 0.971    | 0.985     | 0.952  | 0.968    |

2.2 Task 2 – Knowledge Selection

Once a user turn is determined to be knowledge-seeking by Task 1, Task 2 selects the relevant knowledge snippets $k_i$ from the external knowledge base $K = \{k_1, \ldots, k_n\}$ based on a dialog history $U_t$. While one or more knowledge snippet(s) may be fitting for an answer, only one is considered most relevant and correct in the DSTC9 Track 1 challenge (Kim et al. 2020).

A Factorized Approach We model Task 2 as a knowledge retrieval (or more specifically, document retrieval) problem, i.e., given the query dialog history $U_t$, we retrieve the most relevant knowledge snippet $k_i \in K$ from the set of all knowledge snippets $K$ ranked by a function $f$. In this context, the function $f$ is the probability of selecting a knowledge snippet $k_i \in K$ where $d_i \in D_w$, or $k_i \in e_i$ where $e_i \in_d d_i, d_i \in D_e$ ($e_i \in_d$ and $e_i \in_d$ denote the relations ‘belongs to the domain’ and ‘belongs to the entity’ respectively). Therefore, the selected knowledge snippet $k_i$ is given by:

$$\text{arg max}_{k_i} f(k_i \mid U_t) = \text{arg max}_{k_i \in_d d_i} P(d_i \mid U_t)$$

We propose that we first recognize the possible target domains $d_i \in D_w$ and entities $e_i \in_d d_i$ where $d_i \in D_e$ and estimate the relevance of the domains to the dialog history before choosing the appropriate knowledge snippet, since it can drastically narrow the search space for knowledge snippets. In other words, factorization reduces the problem of Task 2 into three sub-tasks, for each of which models can be trained for target discrimination. Consequently, we have:

$$\text{arg max}_{k_i \in_d d_i \in D'} P(d_i \mid U_t) P(k_i \mid d_i, U_t)$$

where $D' = \{d_i : d_i \in D_w, d_i \in O_{DE} \} \cup \{d_i : e_i \in_d d_i, d_i \in D_e, e_i \in O_{DE} \} (O_{DE}$ refers to the output of Module 1, which is the set of extracted domains and entity candidates). $P(d_i \mid U_t)$ and $P(k_i \mid d_i, U_t)$ are estimated using Modules 2 and 3 respectively. The three modules are described in the following sections.
BERT Backbone. The computation of the factored probabilities \( P(k_i \mid d_i, U_t) \) and \( P(d_i \mid U_t) \) naturally resorts to natural language understanding (NLU) models. We employ BERT (Devlin et al. [2018]) as the NLU backbone and propose KnowledgeFactor (A Factorized Approach to Domain, Entity and Knowledge Selection). Three neural models are developed—the BERT for Domain & Entity Model (BERT-DE) in Module 1, BERT for Domain Model (BERT-D) in Module 2 and BERT for Knowledge Model (BERT-K) in Module 3.

Module 1 - Domain and Entity Selection. We use the heuristics-based surface matching algorithm SMA (described in Section 2.1) to match the possible domains \( d_i \in D_w \) and entities \( e_i \in_d d_i \) where \( d_i \in D_e \).

In view of the high generalization power of neural models, we propose a domain-entity classifier (BERT-DE) to refine the results obtained by SMA.

A dialog history \( U_t \) is concatenated with a domain \( d_i \) (and an entity \( e_i \in_d d_i \), if \( d_i \in D_e \)) as the input to the BERT-DE. For example, \( \text{train} \ (\in \ D_w) \) and \( \text{hotel} \ (\in \ D_e) \) concatenated with \( \text{Autumn House} \) (\( \in_e \text{‘hotel’} \)) are the two kinds of input.

BERT-DE computes the probability that the dialog history \( U_t \) is relevant to each domain \( \in D_w \) or entity \( \in_e d_i \), where \( d_i \in D_e \) and outputs the top-1 result with the highest probability, which is then added to the candidates if it has a different domain than that of the top-1 retrieved by SMA. In the end, we keep at most one entity per domain and finally output \( O_{DE} \) for Module 3.

Module 2 - Domain Probability Estimation. BERT-D is a multi-class domain classifier. Given the concatenation of a dialog history \( U_t \) and a domain \( d_i \) (e.g., \text{hotel}, \text{train}, etc.) as input, it estimates and outputs \( P(d_i \mid U_t) \), the probability that \( U_t \) is relevant to \( d_i \).

We combine the DSTC9 Track 1 Training Set with the MultiWOZ2.2 Data Set (Zang et al. [2020]) to fine-tune the BERT-D on eight domains, i.e., \text{hotel}, \text{restaurant}, \text{train}, \text{taxi}, \text{attraction}, \text{hospital}, \text{police} and \text{bus}, to make the model more generalized and robust.

The BERT-D differs from the domain classifier in Task 1 since we only focus on knowledge-seeking turns whereas Task 1’s model needs to be applied to both knowledge-seeking and non-knowledge-seeking turns. Examples for the two cases are as follows:

Case 1 - Non-knowledge-seeking Turn
User: I am looking for an expensive indian restaurant in the area of centre.
Task 1 Domain Classifier: restaurant.
Task 2 Domain Classifier: N/A (ignore the turn).

Case 2 - Knowledge-seeking Turn
User: Does this hotel offer its guests wifi services?
Task 1 Domain Classifier: hotel.
Task 2 Domain Classifier: hotel.

Module 3 - Knowledge Probability Estimation. The BERT-K is designed to estimate \( P(k_i \mid d_i, U_t) \) for all knowledge snippets \( k_i \) of the domains and entities selected in Module 1. As most of the users’ queries are embedded in the current user turn, the input to the BERT-K is the concatenation of the current user utterance \( U_t \), a domain \( d_i \) and a knowledge snippet \( k_i \) (title & body). For both the current user utterance and the knowledge snippet, any matched entity name is replaced by its domain so that the model only focuses on the semantics of the query but not any information about the entity, which has already been processed by previous modules.

2.3 Task 3 – Knowledge-grounded Response Generation
Task 3 takes a knowledge-integrative approach to generate a system response based on the dialog history \( U_t \) and the top- \( k \) ranking knowledge snippets \( k_i \) based on their confidence values \( p_i \), which are provided by Task 2. We develop an ensemble system Ens-GPT that incorporates two different approaches to deal with the two scenarios (ID and OOD). If the domain of the top knowledge snippet was seen in training then response generation will be conducted as ID and otherwise as OOD. For ID cases, we adopt a Neural Response Generation approach. For OOD cases, we adopt a retrieval-based approach referred to as Neural-Enhanced Response Reconstruction.

Neural Response Generation. Our neural response generation approach GPT2-XL with multi-knowledge snippets (GPT2-XL for short) follows the DSTC9 Track 1 baseline neural generation model in (Kim et al. [2020]) to leverage the large pre-trained language model GPT2. The baseline neural generation model uses the ground truth knowledge snippet and dialog history \( U_t \) as input for fine-tuning GPT2 small, and the ground truth response \( r_{t+1} \) as target. During testing, the baseline model uses knowledge from the top-ranking snippet output by Task 2.

As GPT2 XL has a greater number of parameters to capture more information, we adopt the much larger pre-trained model GPT2 XL other than the GPT2 small used in the baseline model. We should note that the actual correct knowledge snippet may not always rank top in the shortlisted snippets from Task 2, but most of the time they lie within the top 5 retrieved snippets. Hence, we use multiple knowledge snippets in the input, \( n \) in total. For model fine-tuning, besides the ground truth snippet, we also randomly select \( n - 1 \) additional snippets that have the same domain and entity with the ground truth snippet and append them to the input. Correspondingly, we use top- \( n \) snippets from the retrieved top-ranking snippet list from Task 2 in the input for evaluation.

Neural-Enhanced Response Reconstruction. Typical responses may consist of two parts—(i) an informative body which answers to the user’s query; and (ii) a prompt to move the dialog forward. For example:

User: “Does this hotel allow children to stay there?”

Ground Truth Response: “Kids of all ages are welcome as guests of this establishment. Do you want to proceed with the booking?”

Since the knowledge snippets made available are derived from FAQs, the top snippet is used as the body in the response. Therefore, the GPT2-XL Response Reconstruction
Table 4: Evaluation results of knowledge selection task on DSTC9 Track 1 Validation Set for all true knowledge-seeking turns. Line-1 is the reproduced GPT2-Baseline and line-2 is the performance of KnowleDgEFactor.

| Model                      | MRR@5 | R@1  | R@5  |
|----------------------------|-------|------|------|
| Reproduced Baseline        | 0.830 | 0.731| 0.957|
| KnowleDgEFactor            | 0.973 | 0.964| 0.984|

(GPT2-XL-RR) method forms an informative and accurate response by replacing the body of the neural generated response with the top-ranking snippet, while preserving the prompt in the generated response. For example, given:

Top-ranking knowledge snippet: "Children of any age are welcome at The Lensfield Hotel."

GPT2-XL generated response: “Yes, The Lensfield Hotel welcomes children to stay. Should I make the reservation now?”

The GPT2-XL-RR constructs the response as “Children of any age are welcome at The Lensfield Hotel. Should I make the reservation now?”

Ensemble System. To utilize the two approaches above, a decision tree is designed for the ensemble system Ens-GPT. The system first checks if the user query is ID or OOD, which is detected by Task 2 and indicated by the domain of the top-ranking retrieved snippet. For ID cases with available training data, the neural model GPT2-XL is generally well-trained, so it can generate relevant responses to the dialog even when the correct retrieved snippet is not retrieved. Therefore, given ID user queries, GPT2-XL is used for response generation.

On the other hand, if the current user query is OOD, the ensemble’s heuristic will check if the top-ranking knowledge snippet has a sufficiently high confidence value \( p \) (which is empirically set as 5x of the confidence of the second highest ranking knowledge snippet). If this condition is met, implying that the top-ranking snippet is very likely correct, then GPT2-XL-RR is used for response generation. Otherwise, the ensemble method falls back to GPT2-XL, which can extract information from all top-k snippets, rather than only utilizing the single top snippet.

3 Experiments

3.1 Task 1 – Knowledge-seeking Turn Detection

Evaluation Metrics. We use precision, recall and F-Measure as the metrics to evaluate the knowledge-seeking turn detection task (Gunasekara et al. 2020).

Experimental Settings. We use HuggingFace’s implementation (Wolf et al. 2019) of BART (large) model for the Domain and Knowledge Classifier. The models were trained independently with a batch size of 120 and 3704 warmup steps. The models were trained for 4 epochs and the epoch with best performance on validation set was chosen. In our submitted system, we use \( N_{\text{dialog}} = 2 \) with premise and hypothesis templates, i.e., both the system and user response for the Domain Classifier, and \( N_{\text{dialog}} = 1 \) without any templates, i.e., only user response for the Knowledge Classifier. In a later improved knowledge classifier, we use \( N_{\text{dialog}} = 2 \) with premise template. To test the generalizability on OOD user queries, we train and test the baseline and KDEAK on 4 versions of 2 disjoint sets of domains, with 2 domains in each, respectively.

3.2 Task 2 – Knowledge Selection

Evaluation Metrics. The performance of KnowleDgEFactor is measured in terms of standard information retrieval evaluation metrics, including recall and mean reciprocal rank (Kim et al. 2020).

Experimental Settings. The PyTorch implementation of the BERT base model (uncased) in HuggingFace Transformers (Wolf et al. 2019) is utilized. All three models (BERT-DE, BERT-D, BERT-K) are fine-tuned independently with 10 epochs. The number of negative candidates is set as 4, 3 and 8 for BERT-DE, BERT-D and BERT-K. The maximum token lengths of dialogue and knowledge are 256 and 128 for BERT-DE and BERT-D; 128 and 128 for BERT-K.

3.3 Task 3 – Knowledge-grounded Response Generation

Evaluation Metrics. Standard objective evaluation metrics are used for the system-generated response in comparison with the ground truth – BLEU (Papineni et al. 2002), METEOR (Lavie and Agarwal 2007) and ROUGE (Lin and Och 2004) (Lin and Hovy 2003).

Experimental Settings. We fine-tuned the pre-trained GPT2 XL on DSTC9 training set, and the loss function is the standard language modeling objective: cross-entropy between the generated response and the ground truth response. We set the input length limit as 128 tokens (i.e., words) for the dialog history and 256 tokens for the knowledge snippets. This means that we can typically fine-tune with 9 dialog turns and 4 snippets. The model is trained for 3 epochs with a size of 4. The gradient accumulation and gradient clipping with a max norm of 1.0 were performed at every step. The optimizer was Adam and the learning rate was \( e^{-6} \).

To achieve better performance with the generation model, we also compare different numbers of snippets to find the best setting that can provide enough information without introducing too much noise. Table 10 presents the result on validation set with 1 to 5 snippets. GPT2-XL and GPT2-XL-RR are evaluated on the Test Set in isolation and in combination in the ensemble system in Table 8.

4 Results and Analysis

4.1 Task 1 – Knowledge-seeking Turn Detection

Table 3 summarizes the results of Task 1. KDEAK outperforms the baseline on 3 out of 4 versions of the OOD F1-Score evaluations. The Domain Classifier shows 98.7% accuracy on the DSTC9 Track 1 Val Set. Exploiting the pre-trained knowledge and rich hypothesis of the NLI model, KDEAK is more robust against unseen domains compared to non-NLI based GPT-2 baseline. It offers a transparent
Table 5: Evaluation results of knowledge selection task on the DSTC9 Track 1 Test Set. The 1st row is the released results on official GPT2-Baseline and the 2nd row shows KnowleDgEFactor’s performance. The 3rd row shows the results operated on the ground-truth Task 1 prediction to evaluate our system independently.

| Model                  | Source of Task 1 Predictions | MRR@5 | R@1  | R@5  |
|-----------------------|------------------------------|-------|------|------|
| Official GPT2-Baseline | Official GPT2-Baseline       | 0.726 | 0.620| 0.877|
| KnowleDgEFactor       | Task 1 Reproduced GPT2-Baseline | 0.853 | 0.827| 0.896|
| KnowleDgEFactor       | Ground Truth                | 0.903 | 0.867| 0.960|

Table 6: Percentage of the true knowledge-seeking turns with different number of domain and entity candidates retrieved by Module 1.

| #Candidates | Percentage |
|-------------|------------|
|             | Validation Set | Test Set |
| 1           | 84.1        | 74.0     |
| 2           | 13.8        | 23.7     |
| 3           | 1.5         | 2.2      |
| 4           | 0           | 0.1      |
| 5           | -           | 0        |

Table 7: Performance of domain, entity and knowledge selection for top-1 knowledge snippet by KnowleDgEFactor on true knowledge-seeking turns of DSTC9 Track 1 Test Set. A turn is considered for entity accuracy calculation only if the predicted domain is correct and for knowledge accuracy calculation only if both domain and entity are correct.

| True Domain | #Correct (%Correct) |
|-------------|---------------------|
|             | Domain | Entity | Knowledge |
| Hotel       | 567   (98.8) | 545   (96.1) | 513   (94.1) |
| Restaurant  | 599   (98.0) | 577   (96.3) | 554   (96.0) |
| Taxi        | 183   (98.9) | -     | 138   (75.4) |
| Train       | 346   (99.7) | -     | 283   (81.8) |
| Attraction  | 256   (97.0) | 243   (94.9) | 230   (94.7) |

4.2 Task 2 – Knowledge Selection

Strength of a Factorized Approach. The performance improvements on both DSTC9 Track 1 Validation (Table 4) and Test (Table 5) Sets over the baseline model demonstrate the advantage of a factorized approach to knowledge selection. One possible advantage of dividing the computation could be that the domain, entity and knowledge information from the dialogs is disentangled, and consequently the models of the three modules can respectively capture the traits about the three sub-tasks more easily. 84% and 74% of the true knowledge-seeking turns have only one entity or one domain $d_i \in D_w$ retrieved by Module 1 from each dialog of the Validation and Test Sets respectively (Table 6), and over 93% of the top-1 retrieved entity is correct on both data sets, demonstrating the robustness and precision brought by the SMA and the BERT-DE.

Error Analysis. Despite the improved performance over the baseline, there is a noticeable decline of accuracy on the DSTC9 Track 1 Test Set as compared to that on the Validation Set. The drop can be attributed to the inability of Module 3 to recover the correspondence between the current user utterance and the knowledge snippets that are unseen during training. Table 7 records the domain-, entity- and knowledge-level accuracies of the top-1 selected knowledge snippet for all true knowledge-seeking turns. Although most of the selections are over 94% accurate, it is shown that the errors mainly originate from the incorrect knowledge selection on the train and taxi domains, where we find that KnowleDgEFactor sometimes fails to distinguish between similar knowledge snippets. For example, 19 similar errors are found when the user asks about payment under the domain taxi. Figure 2 shows an example of such erroneous instances where the correct knowledge snippet is ranked third. In this example, KnowleDgEFactor associates ‘pay’ in the user query and ‘payments’ in the title of the selected knowledge snippet without attending to the signaling word ‘tip’.

4.3 Task 3 – Knowledge-grounded Response Generation

Empirical results in Table 10 indicate that the use of an appropriate number of additional knowledge snippets (i.e., $n = 4$ in total) tends to result in improved performance compared to exclusive use of the top-ranking snippet. However, when $n \neq 4$ performance degrades.

Table 8 shows that all the methods, namely GPT2-XL and GPT2-XL-RR, as well as their ensemble, outperform the baseline (GPT2-small with a single knowledge snippet).
We note that the approaches to Tasks 1 and 2 evolved to become convergent with some overlapping goals, but they are still different in certain fundamental aspects. In the future, we would like to develop a more streamlined approach, possibly combining Tasks 1 and 2 into a single sub-system.

## 5 Conclusion

We presented a pipeline of KDEAK, KnowleDgEFactor, and Ens-GPT, which achieves task-oriented dialog modeling with unstructured knowledge access, that can respond to users’ request for information lying outside the database but in an external knowledge repository of FAQ-like snippets.

Task 1 (knowledge-seeking turn detection) is accomplished by a subsystem named KDEAK. It formulates the problem as natural language inference and fully utilizes three information sources – dialog history, database, and external knowledge. Domain and entity information determine the candidate pool of information snippets which are ranked based upon relevance to the user’s query. Final classification is based on the source of most relevant information snippet.

Task 2 (knowledge selection) resorts to a 3-module KnowleDgEFactor subsystem formulated as a knowledge/document retrieval problem. It is factorized into the sub-problems of domain and entity selection, domain probability estimation and knowledge probability estimation, which are handled by three modules. Knowledge snippets are ranked using the probabilities computed by the estimates of the modules.

Finally, Task 3 (knowledge-grounded response generation) is performed by Ens-GPT, in which multiple retrieved knowledge snippets are integrated to enrich knowledge and improve robustness of the generated response. The domain of the user query and the confidence of the retrieved snippets are used to determine which way to generate the response.

Automatic evaluation metrics show that the final responses generated from integration of the three subsystems outperform the baseline significantly.

Possible future directions may include extension towards open-domain knowledge-grounded conversations (Gopalakrishnan et al. 2019), enhancing robustness towards recognition errors for speech inputs (Gopalakrishnan et al. 2020) and creating an engaging user experience.

## 6 Acknowledgments

This work is partially supported by the Centre for Perceptual and Interactive Intelligence, a CUHK InnoCentre. We thank Dr. Pengfei Liu, a graduate of the Department of Systems Engineering & Engineering at CUHK, for constructive comments and suggestions.
References

Dagan, I.; Glickman, O.; and Magnini, B. 2005. The PASCAL Recognising Textual Entailment Challenge. In Proceedings of the First International Conference on Machine Learning Challenges: Evaluating Predictive Uncertainty Visual Object Classification, and Recognizing Textual Entailment, MLCW’05, 177–190. Berlin, Heidelberg: Springer-Verlag. ISBN 3540334270. doi:10.1007/11736790_9. URL https://doi.org/10.1007/11736790_9.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 .

Eric, M.; Goel, R.; Paul, S.; Kumar, A.; Sethi, A.; Ku, P.; Goyal, A. K.; Agarwal, S.; Gao, S.; and Hakkani-Tür, D. 2019. MultiWOZ 2.1: A Consolidated Multi-Domain Dialogue Dataset with State Corrections and State Tracking Baselines.

Gopalakrishnan, K.; Hedayatnia, B.; Chen, Q.; Gottardi, A.; Kwatra, S.; Venkatesh, A.; Gabriel, R.; and Hakkani-Tür, D. 2019. Topical-Chat: Towards Knowledge-Grounded Open-Domain Conversations. In Proc. Interspeech 2019, 1891–1895. doi:10.21437/Interspeech.2019-3079. URL http://dx.doi.org/10.21437/Interspeech.2019-3079.

Gopalakrishnan, K.; Hedayatnia, B.; Wang, L.; Liu, Y.; and Hakkani-Tür, D. 2020. Are Neural Open-Domain Dialog Systems Robust to Speech Recognition Errors in the Dialog History? An Empirical Study.

Gunasekara, C.; Kim, S.; D’Haro, L. F.; Rastogi, A.; Chen, Y.-N.; Eric, M.; Hedayatnia, B.; Gopalakrishnan, K.; Liu, Y.; Huang, C.-W.; Hakkani-Tür, D.; Li, J.; Zhu, Q.; Luo, L.; Liden, L.; Huang, K.; Shayandeh, S.; Liang, R.; Peng, B.; Zhang, Z.; Shukla, S.; Huang, M.; Gao, J.; Mehri, S.; Feng, Y.; Gordon, C.; Alavi, S. H.; Traum, D.; Eskenazi, M.; Beirami, A.; Eunjoon, Cho; Crook, P. A.; De, A.; Geramifard, A.; Kottur, S.; Moon, S.; Peddar, S.; and Subba, R. 2020. Overview of the Ninth Dialog System Technology Challenge: DSTC9.

Kim, S.; Eric, M.; Gopalakrishnan, K.; Hedayatnia, B.; Liu, Y.; and Hakkani-Tür, D. 2020. Beyond Domain APIs: Task-oriented Conversational Modeling with Unstructured Knowledge Access. arXiv preprint arXiv:2006.03533 .

Lavie, A.; and Agarwal, A. 2007. METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments. In Proceedings of the second workshop on statistical machine translation, 228–231.

Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; and Zettlemoyer, L. 2019. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension.

Lin, C.-Y.; and Och, F. J. 2004. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04), 605–612.

Papineni, K.; Roukos, S.; Ward, T.; and 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, 311–318.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is all you need. In Advances in neural information processing systems, 5998–6008.

Williams, A.; Nangia, N.; and Bowman, S. 2018. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), 1112–1122. New Orleans, Louisiana: Association for Computational Linguistics. doi:10.18653/v1/N18-1101. URL https://www.aclweb.org/anthology/N18-1101.

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.; Scao, T. L.; Gugger, S.; Drame, M.; Lhoest, Q.; and Rush, A. M. 2019. HuggingFace’s Transformers: State-of-the-art Natural Language Processing. ArXiv abs/1910.03771.

Zang, X.; Rastogi, A.; Sunkara, S.; Gupta, R.; Zhang, J.; and Chen, J. 2020. MultiWOZ 2.2 : A Dialogue Dataset with Additional Annotation Corrections and State Tracking Baselines. In Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI, 109–117. Online: Association for Computational Linguistics. doi:10.18653/v1/2020.nlp4convai-1.13. URL https://www.aclweb.org/anthology/2020.nlp4convai-1.13.