

# Learning Knowledge Bases with Parameters for Task-Oriented Dialogue Systems

Andrea Madotto, Samuel Cahyawijaya, Genta Indra Winata, Yan Xu,  
Zihan Liu, Zhaojiang Lin, Pascale Fung

Center for Artificial Intelligence Research (CAiRE)  
Department of Electronic and Computer Engineering

The Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong

{amadotto, scahyawijaya, giwinata, yxucb, zliucr, zlinao}@connect.ust.hk,  
pascale@ece.ust.hk

## Abstract

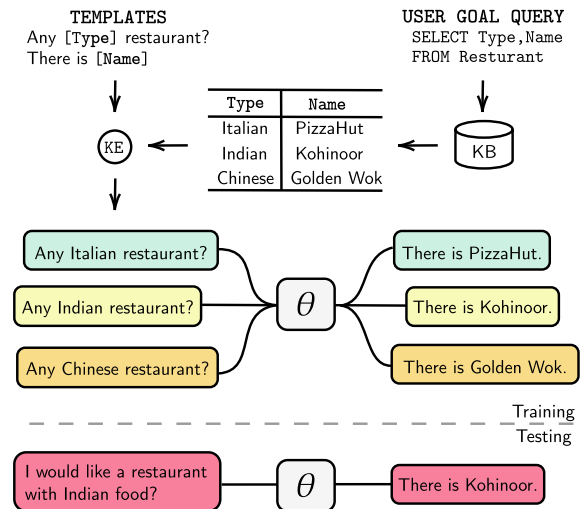
Task-oriented dialogue systems are either modularized with separate dialogue state tracking (DST) and management steps or end-to-end trainable. In either case, the knowledge base (KB) plays an essential role in fulfilling user requests. Modularized systems rely on DST to interact with the KB, which is expensive in terms of annotation and inference time. End-to-end systems use the KB directly as input, but they cannot scale when the KB is larger than a few hundred entries. In this paper, we propose a method to embed the KB, of any size, directly into the model parameters. The resulting model does not require any DST or template responses, nor the KB as input, and it can dynamically update its KB via fine-tuning. We evaluate our solution in five task-oriented dialogue datasets with small, medium, and large KB size. Our experiments show that end-to-end models can effectively embed knowledge bases in their parameters and achieve competitive performance in all evaluated datasets<sup>1</sup>.

## 1 Introduction

Task-oriented dialogue systems are designed to help users achieve predefined goals, such as booking restaurants or movie recommendations via natural language interactions. These systems are deeply connected with external Knowledge Bases (KBs) since the system responses are guided by the output from the KB and the dialogue history.

The current state-of-the-arts (Lei et al., 2018; Zhang et al., 2019a; Mehri et al., 2019; Chen et al., 2019; Peng et al., 2020a; Hosseini-Asl et al., 2020) are end-to-end *pipelined* systems that rely on Dialogue State Tracking (DST) and Speech Act (S-ACT) annotations. Aside from the annotation cost, which is knowingly high (Budzianowski et al.,

<sup>1</sup>Code available in <https://github.com/HLTCHKUST/ke-dialogue>



2018), these pipelined systems must predict a valid DST for querying the KB, execute the query, generate a response template, and finally fulfill it with the retrieved information. The resulting systems are usually overly complicated, and they require multiple steps, including a direct interaction with the KB.

On the other end of the spectrum, there are *end-to-end* trainable models that use both the KB and the dialogue history as input, and they directly generate system responses. Most of the implementations use either the *Gold KB* as input (Eric et al., 2017a; Madotto et al., 2018; Qin et al., 2019, 2020; Banerjee and Khapra, 2019; Neelakantan et al., 2019) or an intermediate API call to retrieve part of the KB (*API+KB*) (Bordes and Weston, 2017; Eric and Manning, 2017; Madotto et al., 2018; Reddy et al., 2019; Wu et al., 2019b). These systems re-

quire at least the DST annotation for generating the API calls or to select the gold KB. Moreover, even with the most advanced transformer architecture (Kitaev et al., 2020; Lample et al., 2019; Child et al., 2019), end-to-end models struggle when the input becomes too large (Neelakantan et al., 2019). For example, in MWOZ (Budzianowski et al., 2018), there are 22K entities just for one of the domains. Interested readers can refer to Appendix C for an overview of different task-oriented methodologies.

On the other hand, Petroni et al. (2019) discovered a simple yet effective way to query factual knowledge from BERT (Devlin et al., 2019). Later on, Roberts et al. (2020) fine-tuned a pre-trained language model, T5 (Raffel et al., 2019), on just question-answers pairs, without letting the model access any external context or knowledge. These results suggest that the actual knowledge is stored in the model parameters. However, in task-oriented dialogue systems, KB entities do not appear in news articles or Wikipedia, e.g., hotel addresses or postcodes, and thus the aforementioned methods cannot be straightforwardly applied, especially when the KB dynamically changes (e.g., weather information).

In this paper, we propose a method to store the KB directly into the model parameters using a novel Knowledge Embedded (KE) approach. The resulting model does not use any DST or template responses, nor a KB as input at the inference time, and it can be used in dynamically changing KBs via fine-tuning. The KE approach consists of a newly defined *user goal query* that generates equivalent KE dialogues from the KB (i.e., table or graph) using minimal annotation effort. Figure 1 shows a high level overview of our approach. To verify the effectiveness of our proposed methodology, we extensively experiment, using both automatic and human metrics, in five task-oriented datasets with small, medium, and large KBs. Our experiments show that end-to-end models can effectively embed knowledge bases in their parameters and achieve competitive performance in all five datasets.

## 2 Methodology

In this section, we formalize the Knowledge Embedded (KE) strategy and the learning algorithm. In Section 2.1, we provide several preliminary definitions used throughout the paper. In Section 2.2, we extend the user goal definition from Schatzmann

et al. (2007) to cover a broad concept that we define as *user goal query*. Then, in Section 2.3, we describe two functions, KE-DELEX and KE-RELEX, used for generating TEMPLATES and KE dialogues, respectively. Finally, in Section 2.4, we describe the Causal Language Model Transformer (Vaswani et al., 2017) used for modeling the dialogue responses.

### 2.1 Preliminary Definition

We define a dataset as a set of dialogues  $\mathcal{D} = \{\mathcal{D}^1, \mathcal{D}^2, \dots, \mathcal{D}^n\}$ . A dialogue  $\mathcal{D}$  is a collection of one or more alternating turns between two speakers, such as  $\mathcal{D} = \{U_1, S_1, \dots, U_t, S_t\}$ , where each  $U$  and  $S$  are sequences of words. Then, we define a table-formatted KB as a set of tuples  $\mathcal{K} = \{(v_1^{a_1}, \dots, v_1^{a_k}), \dots, (v_p^{a_1}, \dots, v_p^{a_k})\}$ , where  $a_1, \dots, a_k \in A$  are the column names of the table,  $v_i^{a_j} \in V_{a_j}$  is the value of tuple  $i$  for the column name  $a_j$ , and  $V_{a_j}$  is a set of possible values for the column name  $a_j$  available in the ontology.

Following the notation in Moon et al. (2019), we define a graph-formatted KB as  $\mathcal{G} = \mathbf{N}_{KG} \times \mathbf{R}_{KG}$ , where  $\mathbf{N}_{KG}$  and  $\mathbf{R}_{KG}$  are the nodes and the relation set, respectively. Then, we define  $\mathbf{N}_r(n)$  as a set of directly connected neighbours of  $n \in \mathbf{N}_{KG}$  by a relation  $r \in \mathbf{R}_{KG}$ . Similarly, we define  $\mathbf{N}_{Rh}(n)$  to be a set of nodes connected to  $n$  via  $h$ -hops with a set of relations  $R$ .

### 2.2 User Goal Query

In task-oriented dialogue systems, the user goal (Schatzmann et al., 2007) for a given dialogue  $\mathcal{D}$  is defined as  $G = (C, R)$ , where  $C$  is a set of constraints that specify the required information, and  $R$  denotes the actual pieces of information of the user desire, (e.g., the name, address, phone number, etc.). The constraint  $C$  is usually expressed by specific values for the attribute, e.g.,  $\{\text{loc=center, price=cheap}\}$ , since there is a one-to-one connection between the user goal and the dialogue. In this paper, we hypothesize that by changing the values of the attributes in  $C$  (e.g.,  $\text{loc=north}$ ) we can generate an equivalent dialogue covering different knowledge.

We leverage the expressive power of *query languages* to describe all the equivalent values that match a particular dialogue, and we name this *User Goal Query*. We use the SQL syntax (Chamberlin and Boyce, 1974) for the table-formatted KB and CYPHER syntax (Webber, 2012) for the graph-formatted KB. Following (Schatzmann et al., 2007),

User Goal Query				TEMPLATE
SELECT	type, poi, distance, address			U: Where is the closest [type]?
FROM	navigation			S: [poi] is [distance] away
GROUP BY	type			U: What is the address?
HAVING	distance = MIN(distance)			S: [poi] is located at [address].
Query Results				KE Dialogue
type	poi	distance	address	U: Where is the closest gas station?
gas station	Valero	5 miles	91 el camino real	S: Valero is 3 miles away
grocery store	safeway	4 miles	452 arcadia pl	U: What is the address?
restaurant	pizzahut	3 miles	915 arbol dr	S: Valero is located at 200 Alester Avenue.

Table 1: A sample of the generated Knowledge Embedded (KE) dialogues. The KE Dialogue are generated by fulfilling the TEMPLATES with the user goal query results.

we define a set of constraints  $C$ , and requirements  $R$  for dialogues with a table-formatted KB, as follows:

$$C = \{\text{OP}(a, v) | a \in A, v \in V_a\}, \quad (1)$$

$$R = \{a | a \in A\} \cup \{a | a \in C\}, \quad (2)$$

where OP is the database operation expressible in an SQL query (e.g., ==, MIN, MAX, SUM, AVG, etc.). The user goal query is then written directly as `SELECT R FROM K WHERE C`.<sup>2</sup>

Similarly, we extend the user goal query definition for datasets with graph-KBs (e.g., OpenDiKG (Moon et al., 2019)). Let us define the  $C$  and  $R$  for dialogues with a graph-formatted KB as:

$$C = \{r | r \in \mathbf{R}_{KG}\}, \quad (3)$$

$$R = \{n | \exists \hat{n} \in \mathbf{N}_{KG}, \hat{n} \in \mathbf{N}_{rh}(n), r \in C\}, \quad (4)$$

where  $h$  is the number of hops. The corresponding user goal query is written directly using CYPHER as `MATCH C RETURN R`, where the node in  $R$  and  $C$  are specified with placeholders (Table A3 in Appendix A). Indeed, a CYPHER query is specified by a graph pattern made of relations in  $\mathbf{R}_{KG}$ . The query results are nodes connected by the specified pattern. In Appendix A.1, we briefly explain the CYPHER query syntax in more details.

### 2.3 Knowledge Embedded (KE)

Given a dialogue  $\mathcal{D}$  and the user goal query, we define two functions: KE-DELEX and KE-RELEX. The KE-DELEX is used to generate the dialogue TEMPLATES, which is a version of  $\mathcal{D}$  where the set of entities related to the user goal query is replaced by their corresponding attribute placeholder. We denote with  $B$  the dictionary that contains the

<sup>2</sup>Notice that we include the attribute specified in  $C$  into  $R$  by overloading the definition of  $\in$

bidirectional mapping between the entities and the corresponding attribute placeholder. Then, the KE-RELEX uses the results from the user goal query to assign new equivalent values to the placeholder in  $B$ . Practically, every TEMPLATE generates as many dialogues as the cardinality of the tuples, or the paths, returned by the user goal query. We denote with  $\mathcal{D}^N$  the newly generated dialogues and we refer to it as KE dialogues.

For example in Table 1, we show a TEMPLATE and user goal query in the SQL syntax, with its resulting output tuples. The dialogue in the example is generated by KE-RELEX using the first tuple, e.g., [Type] is converted into “gas station”, [poi] into “Valero”, and so on.

In the current version of the algorithms, the functions KE-DELEX and KE-RELEX are implemented using string matching. However, they can be implemented using statistical methods; for example, Moon et al. (2019) proposed a model to generate the graph path given a dialogue.

### 2.4 Causal Language Modeling

In this paper, we model the dialogue responses using a Transformer (Vaswani et al., 2017)-based Language Model (LM) (Radford et al., 2019) by using the dialogue history as the prefix in  $\mathcal{D}$  and by autoregressively generating the responses word-by-word  $S_t$  (Wolf et al., 2019a; Zhang et al., 2019b). Let us define the words in  $S_t$  as a set  $\{s_1, \dots, s_n\}$ , then we factorize the language model distribution using the chain rule of probability (Bengio et al., 2003) as:

$$p_\theta(S_t | \mathcal{D}_t) = \prod_i^n p_\theta(s_i | s_{<i}, \mathcal{D}_t), \quad (5)$$

where  $\theta$  are the model parameters and  $\mathcal{D}_t = \{U_1, S_1, \dots, U_t\}$  is the dialogue history. The pa-

Name	Statistics		Seq. Length			KE Statistics	
	#Dial.	#Utt.	Dial.	+GoldKB	+FullKB	#Temp.	#KE-Dial.
<i>bAbI-5</i> (Bordes and Weston, 2017)	3,000	26,326	236	347	10,236	100	55,800
<i>CamRest</i> (Wen et al., 2016)	676	2,744	156	393	1,356	161	32,361
<i>SMD</i> (Eric et al., 2017a)	3,031	15,928	109	435	-	300	2,420
<i>MWOZ</i> <sup>†</sup> (Budzianowski et al., 2018)	2,877	19,870	730	996 <sup>‡</sup>	23,730	527	58,440
<i>OpenDIALKG</i> (Moon et al., 2019)	15,673	91,209	225	292	590,225	11,041	12,593

Table 2: Datasets statistics. **#Temp.** indicates the number of the extracted valid `TEMPLATES`, **#KE-Dial.** indicates the number of generated knowledge-embedded dialogues. We count the maximum input lengths for: dialogue-only (**Dial.**), dialogue with golden KB (**Dial.+GoldKB**), and dialogue with full KB (**Dial.+FullKB**). <sup>‡</sup> as provided by Qin et al. (2020). <sup>†</sup> We consider only single domain dialogues.

parameters in  $\theta$  are trained to minimize the negative log-likelihood over a dataset of dialogues  $\mathcal{D}$ . Formally, we define the  $\mathcal{L}$  as following:

$$\mathcal{L}(\mathcal{D}) = - \sum_k \sum_i^n \log p(s_i^k | s_{<i}^k, \mathcal{D}_t^k), \quad (6)$$

where  $n$  is a maximum response length. Hence, to embed the KB into  $\theta$ , we include the KE dialogues  $\mathcal{D}^N$  in the training set, and we train a Transformer-based Language Model with Equation 6.

### 3 Experiments

In all experiments, if not specifically mentioned, we use the pre-trained GPT2 (small) (Radford et al., 2019) as Causal Language Model (Wolf et al., 2019b). When the dataset has a sufficiently small KB (i.e., less than 1024 tokens), we also fine-tune GPT2 using the KB as input. In Appendix D, we report details about hyperparameters and the implementation details. In Appendix E, we report the data splitting for each dataset.

#### 3.1 Datasets

We use five publicly available multi-turn task-oriented dialog datasets to evaluate our methodology: bAbI-dialogue (bAbI-5) (Bordes and Weston, 2017), Cambridge Restaurant 626 (CamRest) (Wen et al., 2016), In-Car Assistant (SMD) (Eric et al., 2017a), MultiWoZ single (MWOZ) (Budzianowski et al., 2018), and OpenDialKG (Moon et al., 2019). In all datasets, we use the provided split for train/valid/test, except for OpenDialKG where the split was not provided. Dataset statistics are reported in Table 2, including the sequence length of different settings and the number of `TEMPLATES` used for the KE-dialogues.

In all datasets, we use plain text as the input/output sequences instead of their delexicalized

version. This makes the task more challenging, but at the same time more practical because the model produces real entities rather than predefined placeholders, and we do not require additional relexicalization step at the inference time.

#### 3.2 Evaluation Metrics

In bAbI, since it is a synthetic dataset, we use the response and dialogue accuracy (Bordes and Weston, 2017). In CamRest, SMD, MWOZ, and OpenDialKG, we use both the BLEU score (Papineni et al., 2002) and entity F1-score (Eric et al., 2017a). In both CamRest and MWOZ, the existing scorer for the Inform and Success rate (Budzianowski et al., 2018) requires template responses and the predicted DST. Since neither of the two is available for end-to-end models, we implement a plain text scorer for the Inform and Success rate, and we release it, together with our code, for future research. Finally, in OpenDialKG we use the 2-hop neighbors of the entity appearing in the user turn as the gold-reference for the F1-score, which are defined as  $\mathbf{N}_{r2}(n) \forall n \in E(U_t), \exists r \in \mathbf{R}$ , where  $E(U_t)$  are the list of entity nodes appearing in  $U_t$ .

Additionally, we conduct a human evaluation to measure the *Humanness* and *Correctness* of the generated responses. The correctness is computed by counting the ratio of correct entities provided in the generated responses. For the humanness, we use a 4-point Likert Scale, where 1 indicates a non-human-like response, and 4 indicates a very human-like response. All the reported human evaluation results are statistically significant with a p-value < 0.05. Appendix B provides more details of the human evaluation.

#### 3.3 Results

In this section, we describe baselines, training settings, and `KE-DELEX` function in each dataset. Ta-

Model	Test	Test OOV
QRN <sup>1</sup>	99.60 (-)	67.80 (-)
Mem2Seq <sup>2</sup>	97.90 (69.60)	84.50 (2.30)
BoSsNet <sup>3</sup>	97.30 (65.60)	91.70 (18.50)
GLMP <sup>4</sup>	99.20 (88.50)	92.00 (21.70)
GPT2	90.74 (31.00)	70.14 (0.00)
GPT2+KE	<b>99.99 (99.90)</b>	<b>99.01 (94.90)</b>

Table 3: Results on the bAbI dataset. <sup>1</sup>(Seo et al., 2017), <sup>2</sup>(Madotto et al., 2018), <sup>3</sup>(Raghu et al., 2019), <sup>4</sup>(Wu et al., 2019b).

ble 2 summarizes the number of TEMPLATES and KE dialogues generated in each dataset. All generated TEMPLATES are extracted from the training dialogues provided in each dataset. More detailed results for all datasets can be found in Appendix F.

**bAbI-dialog** is a synthetic dataset with five sub-tasks for end-to-end task-oriented models (Bordes and Weston, 2017). Task 1 to 4 is about API calls, refining API calls, recommending options, and providing additional information, respectively. Task 5 is the union of tasks 1-4. Two test-set are provided, one with API combinations appearing in the training set and one with Out-of-Vocabulary APIs. In this paper, we evaluate using task 5 only, in both test sets, by removing all API calls and KB information from the dialogues.

This dataset provides the user goal query directly, and since it is synthetic, the KE-DELEX function is implemented using a string matching. Moreover, we train a GPT2 from scratch using a word-level tokenizer with the bAbI vocabulary. Table 3 compares the performance of GPT2, with and without KE, to existing models that use both API and KB as input. As expected, training GPT2 just on the training dialogues, which covers only 50% of the KB, does not perform well. Instead, by using the KE dialogues in training, GPT2 consistently generates the correct response in both test sets.

**CAMREST** is a human-to-human collected dataset for restaurant booking (Wen et al., 2016). This dataset provides the user goal query, and the KE-DELEX function is implemented using a string matching. We extracted 161 valid TEMPLATES for a total number of 32,361 KE dialogues. Table 4 compares the performance of GPT2, with and without KE, and other models on both automatic and human evaluation. MLMN (Reddy et al., 2019) and BoSsNet (Raghu et al., 2019) use intermediate APIs to select a subset of the KB, where instead

Model	BLEU	F1	Succ.	Hum.	Corr.
KB-Trs <sup>1</sup>	14.80	45.30	-	-	-
MLMN <sup>2</sup>	13.61	54.85	-	-	-
BoSsNet <sup>3</sup>	15.20	43.10	-	-	-
KBRet <sup>4</sup>	<b>18.64</b>	55.76	62.03	3.13	77.33
GPT2	13.58	34.69	30.38	3.42	66.67
GPT2+KB	13.59	50.45	62.03	2.42	70.37
GPT2+KE	18.00	<b>54.85</b>	<b>74.68</b>	<b>3.48</b>	<b>83.50</b>
Human	-	-	86.08	3.60	96.97

Table 4: Results on the CAMREST dataset. <sup>1</sup>(Haihong et al., 2019). <sup>2</sup>(Reddy et al., 2019). <sup>3</sup>(Raghu et al., 2019). We re-evaluate <sup>4</sup>(Qin et al., 2019) using our script that includes postcode as an entity and removes API-calls from F1-count.

KBRet (Qin et al., 2019) uses directly the gold KB. To the best of our knowledge, no models used the entire KB as input, thus we train GPT2 using intermediate API and KB. In general, this setting (GPT2+KB) does not perform as well as similar baselines. This because the KB format is very different from the plain text used for the pre-training. Instead, GPT2+KE is able to achieve better performance than the current state-of-the-art, 1% improvement, with a much shorter input sequence (156 vs 393). From the human evaluation, we notice a significant improvement in favor of GPT2 models, especially GPT2+KE, in both humanness and correctness.

**SMD** is a human-to-human collected dataset (Eric et al., 2017a) with three domains: Navigation, Weather, and Calendar. In this dataset, no user goal query is provided; thus, we manually annotate 100 dialogues per domain from the training set, resulting in as many TEMPLATES. Moreover, to simplify the KE-DELEX function, we also tag the entities in the conversation. Differently from other datasets, the KB dynamically changes in each dialogue and thus requires a KB update operation. To cope with this setting, we propose a fine-tuning approach as follows: given a dialogue KB from the test set, 1) we use the TEMPLATES and the corresponding user goal queries to generate the KE dialogues based on the KB, 2) we fine-tune the GPT2 model with the generated dialogues, and 3) we use the model to generate the response for the considered dialogue sample from the test set. Based on the KB size, for each test sample, we generate, on average, 469/162/6,629 KE dialogues for Navigate/Calendar/Weather, respectively.

Table 5 compares the performance of our method

Model	BLEU	Ent.	Nav.	Wea.	Sch.	Hum.	Cor.
KVRet <sup>1</sup>	13.20	48.00	44.50	53.30	62.90	-	-
MLMN <sup>2</sup>	17.10	55.10	41.30	47.00	68.30	-	-
BoSsNet <sup>3</sup>	8.3	35.9	-	-	-	-	-
Mem2Seq <sup>4</sup>	12.20	33.40	20.00	49.30	32.80	-	-
KBRet <sup>5</sup>	13.90	53.70	54.50	52.20	55.60	-	-
KB-Trs <sup>6</sup>	13.90	37.10	23.30	48.20	51.20	-	-
GLMP <sup>7</sup>	13.90	60.70	54.60	56.50	72.50	-	-
DFF <sup>8</sup>	14.40	<b>62.70</b>	<b>57.90</b>	57.60	<b>73.10</b>	3.28	68.90
GPT2	15.60	39.11	23.41	53.74	52.26	<b>3.49</b>	67.05
GPT2+KB	17.03	58.60	48.37	<b>62.87</b>	72.22	3.47	81.03
GPT2+KE	<b>17.35</b>	<b>59.78</b>	<b>53.53</b>	57.73	<b>72.58</b>	3.44	<b>85.56</b>
Human <sup>1</sup>	13.50	60.70	55.20	61.60	64.30	3.54	97.92

Table 5: Results on the SMD (KVR) dataset. <sup>1</sup>Eric et al. (2017b) <sup>2</sup>(Reddy et al., 2019) <sup>3</sup>(Raghu et al., 2019) <sup>4</sup>(Madotto et al., 2018) <sup>5</sup>(Qin et al., 2019) <sup>6</sup>(Haihong et al., 2019) <sup>7</sup>(Wu et al., 2019b) <sup>8</sup>(Qin et al., 2020)

with existing baselines. Firstly, we notice that GPT2, even without KB, performs better than the existing baselines (Madotto et al., 2018; Haihong et al., 2019; Raghu et al., 2019), suggesting a significant overlapping between the training and test set KBs. As aforementioned, GPT2 with the KB as input does not perform as well as other baselines with a similar setting, except for the Weather domain, where it actually achieves SOTA performance. GPT2 fine-tuned with the KE dialogues performs almost as well as DFF (Qin et al., 2020) in terms of F1-score, but from the human judgments, GPT2-based models perform significantly better both in terms of humanness and correctness.

**MultiWOZ** dataset (Budzianowski et al., 2018) consists of five domains: Train, Attraction, Hotel, Restaurant, and Taxi. Following Qin et al. (2020), we select only the dialogues with a single domain, which is more challenging since less data is available, and we leave the multiple domains per dialogue to future work. This dataset provides both the user goal query and the span annotation for the entities. The KE-DELEX function is implemented using the entity span annotation, although advanced string matching could also work. We extracted 63/116/289/59 TEMPLATES and 3,826/2,495/21,970/30,149 KE dialogues for Attraction/Hotel/Restaurant/Train, respectively. The Taxi domain does not have a KB, since all of its dialogues are booking related.

In Table 6 we compare GPT2 trained with KE dialogues with the current state-of-the-art for pipelined models (DAMD) (Zhang et al., 2019a) and end-to-end models (DFF) (Qin et al., 2020).

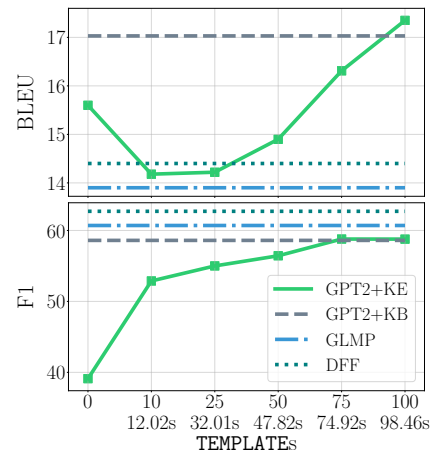


Figure 2: BLEU and F1-Score versus number of TEMPLATES in the SMD dataset.

We re-train DAMD on single domain dialogues, and we use the script provided by the authors to relexicalize the generated templates. We are aware of newly-released models (Hosseini-Asl et al., 2020; Peng et al., 2020a); however, no code was available at submission time for running the results on single domain.

In DFF, we used the provided model to generate the system responses for the human evaluation, but we could not use our scorer to automatically evaluate the Inform, Success, and F1 since no dialogue Id was present in their pre-processed data.<sup>3</sup> Moreover, the authors provided the results in three domains (Attraction, Hotel, Restaurants) for multiple baselines by using the Gold-KB as input.

From our experiments, two points can be highlighted: 1) GPT trained with KE dialogues performs as well as DAMD trained using DST and template responses, in both automatic and human evaluation. Using the original scorer (Budzianowski et al., 2018), DAMD achieved 85.40 Inform and 70.40 Success score, but when the responses are relexicalize and we use our scorer, the results are significantly lower.<sup>4</sup> The human evaluation confirms the correctness of our plain scorer and it shows that the relexicalization process is not a trivial task; 2) Our model achieves a higher BLEU and F1-score that other models trained with gold KB as input, and it achieve a significantly higher correctness compare to DFF. This is easily explainable by the fact that DFF does not

<sup>3</sup>We reproduce their generated responses from <https://github.com/LooperXX/DF-Net>

<sup>4</sup>We properly align the entities to our scorer.

Model	Inform	Success	BLEU	F1	Train	Attraction	Hotel	Rest	Taxi	Human	Correct
Mem2Seq <sup>1</sup>	-	-	6.60	21.62	-	22.00	21.00	22.40	-	-	-
DSR <sup>2</sup>	-	-	9.10	30.00	-	28.00	27.00	33.40	-	-	-
GLMP <sup>3</sup>	-	-	6.90	32.40	-	24.40	28.10	38.40	-	-	-
DFP <sup>4</sup>	-	-	9.40	35.10	-	28.10	30.60	<b>40.90</b>	-	2.65	25.53
GPT2	64.60	51.77	14.33	30.38	23.30	15.11	23.56	25.62	89.76	3.51	55.91
GPT2+KE	<b>72.57</b>	<b>64.16</b>	<b>15.05</b>	<b>39.58</b>	<b>23.79</b>	<b>43.32</b>	<b>33.44</b>	37.10	<b>92.38</b>	<b>3.56</b>	<b>73.38</b>
DAMD*	72.12	61.06	11.48	-	-	-	-	-	-	3.31	67.97
Human	-	-	-	-	-	-	-	-	-	3.66	96.85

Table 6: Results on the MultiWOZ dataset. <sup>1</sup>(Madotto et al., 2018). <sup>2</sup>(Wen et al., 2018). <sup>3</sup>(Wu et al., 2019b). <sup>4</sup>(Qin et al., 2020). \*We evaluate DAMD (Zhang et al., 2019a) with our plain text scorer.

Model	Iter.	BLEU	Prec.	OOV Prec.
GPT2+PATH	-	<b>7.32</b>	<b>86.41</b>	<b>5.55</b>
GPT2	-	4.89	76.85	0.66
GPT2+KE	3K	<b>5.04</b>	79.14	1.01
GPT2+KE	6K	5.00	78.87	1.40
GPT2+KE	9K	4.72	<b>79.41</b>	1.65
GPT2+KE	12K	4.64	78.59	<b>2.11</b>

Table 7: Results on the OpenDialKG dataset. PATH represents the model with the correct nodes and relations provided from the dataset.

issue booking API and thus it constantly mistakes the booking results. In appendix H, we show how our model handles the booking API.

**OpenDialKG** is a human-to-human collected dataset (Moon et al., 2019) consisting of four domains: Music, Sport, Book, and Movie. No official split is provided and thus we randomly split the dataset in 80/10/10 for the train/valid/test, respectively. The dataset provides a large knowledge graph with 100K entities and 1.1M relations, and the annotated entity path that connects  $U_t$  and  $S_t$ . The graph relations in the annotated path are the user goal query defined in Equation 4, but after a careful analysis, we discover that the annotation is incomplete in most of the dialogues. Therefore, we decided to automatically generate the user goal queries using string matching and the CYPHER query language.<sup>5</sup> This process generates 11K possible TEMPLATES, which, if used over the user goal query output, generate over a billion KE dialogues. This is because the knowledge graph is large, and each user goal query returns a large number of equivalent entities. To overcome this issue, 1) we select a subset of the knowledge graph, 5,691 entities, and 39,728 relations, which covers most of the test set entities, and 2) we iteratively gener-

<sup>5</sup>More details in Appendix A.1

ate dialogues by sampling TEMPLATES and using KE-RELEX over the sampled query results.

Table 7 compares a GPT2 trained with the provided gold path as input with a GPT2 trained on an increasing number of dialogues generated by the iterative procedure. We observe that by increasing the number of iterations, thus the number of KE dialogues, the entity F1-score increases, especially for OOV entities, but at the same time, the BLEU score decreases. After a careful qualitative analysis, we notice that the string matching algorithm used for extracting the user goal queries generate noisy and incomplete TEMPLATES, and thus most of the KE dialogues have imprecise knowledge. We leave the annotation of the user goal queries and the human evaluation to the future work.

## 4 Analysis and Discussions

**Templates vs. Performance** In all experiments, we show that given the generated KE dialogues, the model learns to embed the KB into its parameters. However, the user goal query still requires human annotations; thus, we want to analyze the effect of using increasingly less TEMPLATES in KE. For instance, in Figure 2, we report the number of TEMPLATES used for fine-tuning versus the BLEU score and the entity F1-score in the SMD dataset. In general, we observe that more TEMPLATES increase significantly both the F1 and BLEU score. Especially, we observe that BLEU score linearly increase with the number of TEMPLATES used in training, suggesting that a more diverse and fluent generation can be achieved using more TEMPLATES. In Appendix F, we report the same analysis in each datasets, where we observe a similar trend.

**Limitation & Dynamic KB** Throughout our experiments, we identify two major limitations: noisy KE dialogues generation and fine-tuning time for

dynamic KBs. Although the proposed KE results successfully embed the KB into the model parameters, the generated KE dialogues are sometimes noisy. For example, the KE-DELEX function converts, “i want to find an expensive restaurant...” into a TEMPLATE “i want to find an [price-range] restaurant...”. Then the KE-RELEX can generate “i want to find a cheap restaurant...”, which has a clear grammar mistake. This type of error does not happen often, and we notice that GPT2 is robust to this kind of noisy input. In future work, we propose to improve the robustness and fluency of our model using different regularization losses. Moreover, in the case of dynamic KBs a substantial fine-tuning cost is required for updating the KB. Figure 2 shows the average time-per-epoch spent for fine-tuning in SMD. In future work, we propose to study both a meta-learning (Finn et al., 2017) strategy for quick fine-tuning and continual learning approach for updating the KB while retaining the previous existing knowledge.

## 5 Related Work

**Dialogue Systems** are categorized (Gao et al., 2018) into chit-chat (Vinyals and Le, 2015; Serban et al., 2016) and task-oriented (Williams and Young, 2007; Young et al., 2013); in this paper we focus on the latter. Task-oriented dialogue systems are further classified into: modularized (Levin et al., 2000; Hori et al., 2009; Lee et al., 2009), retrieval (Henderson et al., 2019; Wu et al., 2020) end-to-end (Bordes and Weston, 2017; Eric et al., 2017a; Eric and Manning, 2017; Reddy et al., 2019; Madotto et al., 2018; Wu et al., 2019b; Madotto et al., 2020a; Neelakantan et al., 2019; Qin et al., 2019, 2020; Raghu et al., 2019; Haihong et al., 2019; He et al., 2020) and hybrid (Shu et al., 2018; Lei et al., 2018; Zhang et al., 2019a; Mehri et al., 2019; Chen et al., 2019; Peng et al., 2020a; Ham et al., 2020; Hosseini-Asl et al., 2020; Le et al., 2020; Lin et al., 2020). To the best of our knowledge, these methods use either DST/S-ACT annotations, template responses, or all/partial KB as the input to the model, where instead we only use the dialogue history.

Recently, several task-oriented dialogue models are introduced to tackle the resource scarcity challenges in target domains (Bapna et al., 2017; Shah et al., 2019; Wu et al., 2019a; Liu et al., 2020) and target languages (Mrkšić et al., 2017; Schuster et al., 2019; Chen et al., 2018; Liu et al., 2019b),

and large pre-trained language models are shown to possess the capability to quickly adapt to task-oriented dialogue tasks by using only a few data samples (Peng et al., 2020b; Madotto et al., 2020b; Wu et al., 2020).

**Data Augmentation** is a widely used technique to improve both robustness and performance (Guo et al., 2019; Yang et al., 2020). Task-oriented dialogue systems have been explored to improve DST (Song et al., 2020; Yoo et al., 2020; Campagna et al., 2020), Natural Language Understanding (NLU) (Peng et al., 2020c), intent classification (Kumar et al., 2019) and hybrid end-to-end systems (Zhang et al., 2019a; Rastogi et al., 2019). These data augmentation methods aim to improve the final performance of the given task, e.g., zero-shot performance, template response, etc., where instead, our proposed approach aims to store the KB into the model parameters.

**Agenda-Based User Simulation** builds an interactive system that models the user turns (Schatzmann et al., 2007) rather than the system. User simulators are designed to cover all possible user queries while keeping a diverse and fluent user interaction. This enables models to learn a better dialogue policy via interaction (Asri et al., 2016; Li et al., 2017; Wu et al., 2019c; Peng et al., 2018), and it is especially useful in scenarios in where few or no data is available (Liu and Lane, 2017; Liu et al., 2017; Shah et al., 2018; Kreyssig et al., 2018; Li et al., 2020). In our work, instead, we use all the possible user goal queries to generate dialogues directly, instead of creating a reinforcement learning loop to train the model.

**Language Models as Knowledge Bases** has been used for encoding common sense knowledge into transformers (Bosselut et al., 2019; Liu et al., 2019a; Xiong et al., 2019; Wang et al., 2020, 2019). (Guan et al., 2020) improved story generation by training a Language Model with knowledge triples converted into sentences using predefined templates (Levy et al., 2017). Differently, we extract templates from real data, and we aim to store the KB into the models parameters to be able to extract knowledge directly, instead of improving common sense generation. Moreover, several studies tried to extract (Petroni et al., 2019; Kassner and Schütze, 2019; Petroni et al., 2020) or use (Roberts et al., 2020) large pre-trained models, e.g. BERT (Devlin et al., 2019), as knowledge bases.



## 6 Conclusion

In this paper, we propose to learn the KB directly into the model parameters using a novel Knowledge Embedded approach, that is fundamentally different from giving the KB as input or using the DST for querying the KB. We demonstrate that our approach is scalable to different KB sizes and it can be used with dynamically changing KBs via fine-tuning. Automatic and human evaluations confirm that models with embedded KBs achieve competitive performance in all evaluated datasets. Finally we show, for the first time, that end-to-end models can perform as well as pipelined modularized systems (Zhang et al., 2019a) in the MWoZ single domain dataset.

## Acknowledgements

This work has been partially funded by MRP/055/18 of the Innovation Technology Commission, The Hong Kong SAR Government.

## References

- Layla El Asri, Jing He, and Kaheer Suleman. 2016. A sequence-to-sequence model for user simulation in spoken dialogue systems. *arXiv preprint arXiv:1607.00070*.
- Suman Banerjee and Mitesh M Khapra. 2019. Graph convolutional network with sequential attention for goal-oriented dialogue systems. *Transactions of the Association for Computational Linguistics*, 7:485–500.
- Ankur Bapna, Gokhan Tür, Dilek Hakkani-Tür, and Larry Heck. 2017. Towards zero-shot frame semantic parsing for domain scaling. *Proc. Interspeech 2017*, pages 2476–2480.
- Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155.
- Antoine Bordes and Jason Weston. 2017. Learning end-to-end goal-oriented dialog. *International Conference on Learning Representations*, abs/1605.07683.
- Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. Comet: Commonsense transformers for knowledge graph construction. In *Association for Computational Linguistics (ACL)*.
- Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic. 2018. Multiwoz-a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026.
- Giovanni Campagna, Agata Foryciarz, Mehrad Moradshahi, and Monica S Lam. 2020. Zero-shot transfer learning with synthesized data for multi-domain dialogue state tracking. *arXiv preprint arXiv:2005.00891*.
- Donald D Chamberlin and Raymond F Boyce. 1974. Sequel: A structured english query language. In *Proceedings of the 1974 ACM SIGFIDET (now SIGMOD) workshop on Data description, access and control*, pages 249–264.
- Wenhu Chen, Jianshu Chen, Pengda Qin, Xifeng Yan, and William Yang Wang. 2019. Semantically conditioned dialog response generation via hierarchical disentangled self-attention. *arXiv preprint arXiv:1905.12866*.
- Wenhu Chen, Jianshu Chen, Yu Su, Xin Wang, Dong Yu, Xifeng Yan, and William Yang Wang. 2018. Xlnbt: A cross-lingual neural belief tracking framework. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 414–424.
- Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. 2019. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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)*, pages 4171–4186.
- Mihail Eric, Lakshmi Krishnan, Francois Charette, and Christopher D. Manning. 2017a. Key-value retrieval networks for task-oriented dialogue. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 37–49. Association for Computational Linguistics.
- Mihail Eric, Lakshmi Krishnan, Francois Charette, and Christopher D Manning. 2017b. Key-value retrieval networks for task-oriented dialogue. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 37–49.
- Mihail Eric and Christopher Manning. 2017. A copy-augmented sequence-to-sequence architecture gives good performance on task-oriented dialogue. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 468–473, Valencia, Spain. Association for Computational Linguistics.

- Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135. JMLR. org.
- Jianfeng Gao, Michel Galley, and Lihong Li. 2018. Neural approaches to conversational ai. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1371–1374. ACM.
- Jian Guan, Fei Huang, Zhihao Zhao, Xiaoyan Zhu, and Minlie Huang. 2020. A knowledge-enhanced pre-training model for commonsense story generation. *arXiv preprint arXiv:2001.05139*.
- Hongyu Guo, Yongyi Mao, and Richong Zhang. 2019. Augmenting data with mixup for sentence classification: An empirical study. *arXiv preprint arXiv:1905.08941*.
- E Haihong, Wenjing Zhang, and Meina Song. 2019. Kb-transformer: Incorporating knowledge into end-to-end task-oriented dialog systems. In *2019 15th International Conference on Semantics, Knowledge and Grids (SKG)*, pages 44–48. IEEE.
- Donghoon Ham, Jeong-Gwan Lee, Youngsoo Jang, and Kee-Eung Kim. 2020. End-to-end neural pipeline for goal-oriented dialogue systems using GPT-2. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 583–592, Online. Association for Computational Linguistics.
- Z. He, Y. He, Q. Wu, and J. Chen. 2020. Fg2seq: Effectively encoding knowledge for end-to-end task-oriented dialog. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8029–8033.
- Matthew Henderson, Ivan Vulić, Daniela Gerz, Inigo Casanueva, Paweł Budzianowski, Sam Coope, Georgios Spithourakis, Tsung-Hsien Wen, Nikola Mrkšić, and Pei-Hao Su. 2019. Training neural response selection for task-oriented dialogue systems. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5392–5404.
- Chiori Hori, Kiyonori Ohtake, Teruhisa Misu, Hideki Kashioka, and Satoshi Nakamura. 2009. Statistical dialog management applied to wfst-based dialog systems. In *IEEE International Conference on Acoustics, Speech and Signal Processing, 2009. ICASSP 2009.*, pages 4793–4796. IEEE.
- Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. 2020. A simple language model for task-oriented dialogue. *arXiv preprint arXiv:2005.00796*.
- Nora Kassner and Hinrich Schütze. 2019. Negated lama: Birds cannot fly. *arXiv preprint arXiv:1911.03343*.
- Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*.
- Florian Kreyszig, Inigo Casanueva, Paweł Budzianowski, and Milica Gasic. 2018. Neural user simulation for corpus-based policy optimisation for spoken dialogue systems. *arXiv preprint arXiv:1805.06966*.
- Varun Kumar, Hadrien Glaude, Cyprien de Lichy, and William Campbell. 2019. A closer look at feature space data augmentation for few-shot intent classification. *arXiv preprint arXiv:1910.04176*.
- Guillaume Lample, Alexandre Sablayrolles, Marc’Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2019. Large memory layers with product keys. In *Advances in Neural Information Processing Systems*, pages 8546–8557.
- Hung Le, Doyen Sahoo, Chenghao Liu, Nancy F Chen, and Steven CH Hoi. 2020. Uniconv: A unified conversational neural architecture for multi-domain task-oriented dialogues. *arXiv preprint arXiv:2004.14307*.
- Cheongjae Lee, Sangkeun Jung, Seokhwan Kim, and Gary Geunbae Lee. 2009. Example-based dialog modeling for practical multi-domain dialog system. *Speech Communication*, 51(5):466–484.
- Wenqiang Lei, Xisen Jin, Min-Yen Kan, Zhaochun Ren, Xiangnan He, and Dawei Yin. 2018. Sequicity: Simplifying task-oriented dialogue systems with single sequence-to-sequence architectures. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1437–1447.
- Esther Levin, Roberto Pieraccini, and Wieland Eckert. 2000. A stochastic model of human-machine interaction for learning dialog strategies. *IEEE Transactions on speech and audio processing*, 8(1):11–23.
- Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 333–342.
- Xiujun Li, Yun-Nung Chen, Lihong Li, Jianfeng Gao, and Asli Celikyilmaz. 2017. End-to-end task-completion neural dialogue systems. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 733–743.
- Ziming Li, Sungjin Lee, Baolin Peng, Jinchao Li, Shahin Shayandeh, and Jianfeng Gao. 2020. Guided dialog policy learning without adversarial learning in the loop. *arXiv preprint arXiv:2004.03267*.
- Zhaojiang Lin, Andrea Madotto, Genta Indra Winata, and Pascale Fung. 2020. Mintl: Minimalist transfer learning for task-oriented dialogue systems. *arXiv preprint arXiv:2009.12005*.

- Bing Liu and Ian Lane. 2017. Iterative policy learning in end-to-end trainable task-oriented neural dialog models. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 482–489. IEEE.
- Bing Liu, Gokhan Tur, Dilek Hakkani-Tur, Pararth Shah, and Larry Heck. 2017. End-to-end optimization of task-oriented dialogue model with deep reinforcement learning. *arXiv preprint arXiv:1711.10712*.
- Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. 2019a. K-bert: Enabling language representation with knowledge graph. *arXiv preprint arXiv:1909.07606*.
- Zihan Liu, Jamin Shin, Yan Xu, Genta Indra Winata, Peng Xu, Andrea Madotto, and Pascale Fung. 2019b. Zero-shot cross-lingual dialogue systems with transferable latent variables. 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)*, pages 1297–1303.
- Zihan Liu, Genta Indra Winata, Peng Xu, and Pascale Fung. 2020. Coach: A coarse-to-fine approach for cross-domain slot filling. *arXiv preprint arXiv:2004.11727*.
- Andrea Madotto, Zhaojiang Lin, Chien-Sheng Wu, Jamin Shin, and Pascale Fung. 2020a. Attention over parameters for dialogue systems. *arXiv preprint arXiv:2001.01871*.
- Andrea Madotto, Zihan Liu, Zhaojiang Lin, and Pascale Fung. 2020b. Language models as few-shot learner for task-oriented dialogue systems. *arXiv e-prints*, pages arXiv–2008.
- Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. 2018. Mem2seq: Effectively incorporating knowledge bases into end-to-end task-oriented dialog systems. *arXiv preprint arXiv:1804.08217*.
- Shikib Mehri, Tejas Srinivasan, and Maxine Eskenazi. 2019. Structured fusion networks for dialog. *arXiv preprint arXiv:1907.10016*.
- Seungwhan Moon, Pararth Shah, Anuj Kumar, and Rajen Subba. 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*, pages 845–854.
- Nikola Mrkšić, Ivan Vulić, Diarmuid Ó Séaghdha, Ira Leviant, Roi Reichart, Milica Gašić, Anna Korhonen, and Steve Young. 2017. Semantic specialization of distributional word vector spaces using monolingual and cross-lingual constraints. *Transactions of the Association for Computational Linguistics*, 5:309–324.
- Arvind Neelakantan, Semih Yavuz, Sharan Narang, Vishaal Prasad, Ben Goodrich, Daniel Duckworth, Chinnadhurai Sankar, and Xifeng Yan. 2019. Neural assistant: Joint action prediction, response generation, and latent knowledge reasoning. *arXiv preprint arXiv:1910.14613*.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.
- Baolin Peng, Chunyuan Li, Jinchao Li, Shahin Shayan-deh, Lars Liden, and Jianfeng Gao. 2020a. Soloist: Few-shot task-oriented dialog with a single pre-trained auto-regressive model. *arXiv preprint arXiv:2005.05298*.
- Baolin Peng, Xiujun Li, Jianfeng Gao, Jingjing Liu, and Kam-Fai Wong. 2018. Deep dyna-q: Integrating planning for task-completion dialogue policy learning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2182–2192.
- Baolin Peng, Chenguang Zhu, Chunyuan Li, Xiujun Li, Jinchao Li, Michael Zeng, and Jianfeng Gao. 2020b. Few-shot natural language generation for task-oriented dialog. *arXiv preprint arXiv:2002.12328*.
- Baolin Peng, Chenguang Zhu, Michael Zeng, and Jianfeng Gao. 2020c. Data augmentation for spoken language understanding via pretrained models. *arXiv preprint arXiv:2004.13952*.
- Fabio Petroni, Patrick Lewis, Aleksandra Piktus, Tim Rocktäschel, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. 2020. How context affects language models’ factual predictions. *arXiv preprint arXiv:2005.04611*.
- Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? 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)*, pages 2463–2473.
- Libo Qin, Yijia Liu, Wanxiang Che, Haoyang Wen, Yangming Li, and Ting Liu. 2019. Entity-consistent end-to-end task-oriented dialogue system with kb retriever. *arXiv preprint arXiv:1909.06762*.
- Libo Qin, Xiao Xu, Wanxiang Che, Yue Zhang, and Ting Liu. 2020. Dynamic fusion network for multi-domain end-to-end task-oriented dialog. *arXiv preprint arXiv:2004.11019*.

- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.
- Dinesh Raghu, Nikhil Gupta, et al. 2019. Disentangling language and knowledge in task-oriented dialogs. 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)*, pages 1239–1255.
- Pushpendre Rastogi, Arpit Gupta, Tongfei Chen, and Lambert Mathias. 2019. Scaling multi-domain dialogue state tracking via query reformulation. *arXiv preprint arXiv:1903.05164*.
- Revanth Gangi Reddy, Danish Contractor, Dinesh Raghu, and Sachindra Joshi. 2019. Multi-level memory for task oriented dialogs. 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)*, pages 3744–3754.
- Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? *arXiv preprint arXiv:2002.08910*.
- Jost Schatzmann, Blaise Thomson, Karl Weilhammer, Hui Ye, and Steve Young. 2007. Agenda-based user simulation for bootstrapping a pomdp dialogue system. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, pages 149–152. Association for Computational Linguistics.
- Sebastian Schuster, Sonal Gupta, Rushin Shah, and Mike Lewis. 2019. Cross-lingual transfer learning for multilingual task oriented dialog. 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)*, pages 3795–3805.
- Minjoon Seo, Sewon Min, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Query-reduction networks for question answering. In *Proceedings of the International Conference on Learning Representations (ICLR)*.
- Iulian Vlad Serban, Ryan Lowe, Laurent Charlin, and Joelle Pineau. 2016. Generative deep neural networks for dialogue: A short review. *arXiv preprint arXiv:1611.06216*.
- Darsh Shah, Raghav Gupta, Amir Fayazi, and Dilek Hakkani-Tur. 2019. Robust zero-shot cross-domain slot filling with example values. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5484–5490.
- Pararth Shah, Dilek Hakkani-Tur, Bing Liu, and Gokhan Tur. 2018. Bootstrapping a neural conversational agent with dialogue self-play, crowdsourcing and on-line reinforcement learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*, pages 41–51.
- Lei Shu, Piero Molino, Mahdi Namazifar, Bing Liu, Hu Xu, Huaixiu Zheng, and Gokhan Tur. 2018. Incorporating the structure of the belief state in end-to-end task-oriented dialogue systems. In *2nd Workshop on Conversational AI at Neural Information Processing Systems*, volume 32.
- Xiaohui Song, Liangjun Zang, Yipeng Su, Xing Wu, Jizhong Han, and Songlin Hu. 2020. Data augmentation for copy-mechanism in dialogue state tracking. *arXiv preprint arXiv:2002.09634*.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.
- Oriol Vinyals and Quoc V Le. 2015. A neural conversational model. *arXiv preprint arXiv:1506.05869*.
- Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Cuihong Cao, Daxin Jiang, Ming Zhou, et al. 2020. K-adapter: Infusing knowledge into pre-trained models with adapters. *arXiv preprint arXiv:2002.01808*.
- Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhiyuan Liu, Juanzi Li, and Jian Tang. 2019. Kepler: A unified model for knowledge embedding and pre-trained language representation. *arXiv preprint arXiv:1911.06136*.
- Jim Webber. 2012. A programmatic introduction to neo4j. In *Proceedings of the 3rd annual conference on Systems, programming, and applications: software for humanity*, pages 217–218.
- Haoyang Wen, Yijia Liu, Wanxiang Che, Libo Qin, and Ting Liu. 2018. Sequence-to-sequence learning for task-oriented dialogue with dialogue state representation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3781–3792.
- Tsung-Hsien Wen, David Vandyke, Nikola Mrksic, Milica Gasic, Lina M Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2016. A network-based end-to-end trainable task-oriented dialogue system. *arXiv preprint arXiv:1604.04562*.

- Jason D Williams and Steve Young. 2007. Partially observable markov decision processes for spoken dialog systems. *Computer Speech & Language*, 21(2):393–422.
- Thomas Wolf, Victor Sanh, Julien Chaumond, and Clement Delangue. 2019a. [Transfertransfo: A transfer learning approach for neural network based conversational agents](#). *CoRR*, abs/1901.08149.
- Thomas Wolf, Victor Sanh, Julien Chaumond, and Clement Delangue. 2019b. Transfertransfo: A transfer learning approach for neural network based conversational agents. *arXiv preprint arXiv:1901.08149*.
- Chien-Sheng Wu, Steven Hoi, Richard Socher, and Caiming Xiong. 2020. Tod-bert: Pre-trained natural language understanding for task-oriented dialogues. *arXiv preprint arXiv:2004.06871*.
- Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. 2019a. Transferable multi-domain state generator for task-oriented dialogue systems. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 808–819.
- Chien-Sheng Wu, Richard Socher, and Caiming Xiong. 2019b. Global-to-local memory pointer networks for task-oriented dialogue. In *Proceedings of the International Conference on Learning Representations (ICLR)*.
- Yuxin Wu, Xiujun Li, Jingjing Liu, Jianfeng Gao, and Yiming Yang. 2019c. Switch-based active deep dyna-q: Efficient adaptive planning for task-completion dialogue policy learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7289–7296.
- Wenhan Xiong, Jingfei Du, William Yang Wang, and Veselin Stoyanov. 2019. Pretrained encyclopedia: Weakly supervised knowledge-pretrained language model. *arXiv preprint arXiv:1912.09637*.
- Yiben Yang, Chaitanya Malaviya, Jared Fernandez, Swabha Swayamdipta, Ronan Le Bras, Ji-Ping Wang, Chandra Bhagavatula, Yejin Choi, and Doug Downey. 2020. G-daug: Generative data augmentation for commonsense reasoning. *arXiv preprint arXiv:2004.11546*.
- Kang Min Yoo, Hanbit Lee, Franck Dernoncourt, Trung Bui, Walter Chang, and Sang-goo Lee. 2020. Variational hierarchical dialog autoencoder for dialogue state tracking data augmentation. *arXiv preprint arXiv:2001.08604*.
- Steve Young, Milica Gašić, Blaise Thomson, and Jason D Williams. 2013. Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179.
- Yichi Zhang, Zhijian Ou, and Zhou Yu. 2019a. Task-oriented dialog systems that consider multiple appropriate responses under the same context. *arXiv preprint arXiv:1911.10484*.
- Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2019b. Dialogpt: Large-scale generative pre-training for conversational response generation. *arXiv preprint arXiv:1911.00536*.

## A Knowledge Embedded

We provide intuitive samples of our Knowledge Embedded approach in different datasets. Table A1 and Table A2 shows the user goal query in form of SQL syntax for tabular-formatted KB and how the KE-DELEX generate TEMPLATES. Similarly Table A3 shows the user goal query in CYPHER syntax for graph-formatted KB and how the KE-DELEX generates TEMPLATES. We further discuss the detail of the KE-DELEX for OpenDialKG in the following section.

### A.1 OpenDialKG Knowledge Embedded

In OpenDialKG, we divide the KE-DELEX process into three steps: string matching, spanning tree, and dialogue generation. We perform string matching using cased letters, and we only select the entities with a minimum length of five characters to reduce the detection of false entities. To handle overlapping sequences, such as “The Dark” and “The Dark Knight” in “I enjoy watching The Dark Knight”, we perform a further filtering in each turn and we take the longest string when there is an overlapping between two or more entities.

**String Matching Process** We extract a set of entities that from in the dialogue based on the nodes in the graph. This set of entities are defined as the  $R$  of a user goal. To complete the user goal, we need to find the constraint  $C$ . This can be done by generating a spanning tree from the Knowledge Graph between all entities in  $R$ .

**Spanning Tree** We get all the relations and intermediary nodes between each pair of nodes in  $R$ . The collected relations are what we defined as constraint  $C$  of the user goal. With the given  $R$  and  $C$ , we can build a CYPHER query in form of MATCH  $C$  RETURN  $R$  as mentioned in the Methodology.

**Dialogue Generation** We use the CYPHER query to retrieve the equivalent nodes for the dialogue using neo4j, a graph database which sup-

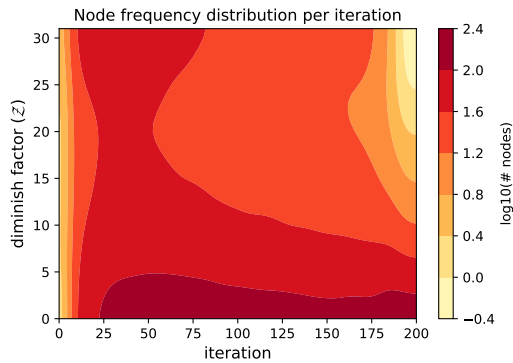


Figure A1: Distribution of #nodes over  $\mathcal{Z}$  and iteration.

ports diverse functionality for graph retrieval and manipulation. An example of our query generation is shown in Table A3. To ensure diversity of the dialogue generation, we set up a diminishing factor  $\mathcal{Z}$  on each node, to restrict the access to the same node over time. We initialize  $\mathcal{Z}$  with the number of edges on each node, and we decremented  $\mathcal{Z}$  each time the node is used for the generation. In order to constraint the query with the limiting factor  $\mathcal{Z}$ , we expand the CYPHER query into `MATCH C WHERE  $\mathcal{Z}_n > 0 \forall n \in \{C, R\}$  RETURN R`. We iteratively generate dialogues by sampling `TEMPLATES`. For each iteration, we randomly sampled 200 `TEMPLATES` and use `KE-RELEX` to generate the dialogues. To check the diversity of the entity in the generated dialogues, we measure the number of nodes per  $\mathcal{Z}$  per iteration. As shown in Figure A1, the nodes with high  $\mathcal{Z}$  is reduced over iteration and on each iteration, more and more nodes reach  $\mathcal{Z} = 0$ , which ensure that the entity selected for the generation of the same `TEMPLATE` would include a different set of entities.

## B Human Evaluation

In this section, we show the annotators instructions used the for the human evaluation.

### B.1 Instructions for Humanness Evaluation

**Overview** In this task, you will be given a dialogue and a response, and you have to provide a rating of the response from 1 to 4 to indicate how human-like is the response. For instance, 4 means that the response is a very natural human response, and 1 indicates the response is obviously not a human-generated response.

**Steps** The steps of the humanness evaluation are as following:

- There is a pre-filled columns with the dialogue history and a second column filled with the response text.
- There is 1 blank humanness column where you can put rating from 1 to 4, indicating how human-like is the response: 4 indicates the response is a very natural human response and 1 indicates the response is obviously not a human-generated response.
- 1. Read the dialogue from the first column.
- 2. Read the response from the second column.
- 3. Rate how human-like is the response and fill the humanness rating on the third column.

### B.2 Instructions for Correctness Evaluation

**Overview** In this task, you will be given a KB, a dialogue history, and a response, and you have to provide a number of entity appearing in the KB and present in the response. You then need to check whether each of the entity is correct given the dialogue history, and the provided KB.

**Steps** The steps of the correctness evaluation are as following:

- There are 3 pre-filled columns, the first column is the ID to the KB, if the KB is dynamic else -1, the second column contains the dialogue history of the conversation, and the third column contains the response.
- There is 2 blank column, the first column (`num_entity`) is where you can put the number of entities existing in the response text and second column (`correct_entity`) is where you can put the number of correct entities based on the dialogue history and the KB.
- Another file for the KB is also provided in separate file named `KB.txt`
- 1. Read the dialogue history and the response from the second and third column.
- 2. Count how many entities on the response text that appears in the KB.
- 3. Find all the possible entities in the KB from the given the response on dialogue history and response and fill the `num_entity` column.

### SMD

U: Where is the closest gas station?  
 S: Valero is 3 miles away.  
 U: What is the address?  
 S: Valero is located at 200 Alester Avenue.

---

**SELECT** type, poi, dist, addr  
**FROM** navigation  
**GROUP BY** type  
**HAVING** dist = MIN(dist)

Table A1: Example of user goal query from SMD (Eric et al., 2017a). The requested information  $R$  are the attribute  $type$ ,  $poi$ ,  $dist$ , and  $addr$ , with the constrain  $C$  using as OP: GROUP BY, HAVING and MIN. The dialogues is filled with an instance from the query results, with the entities marked with the corresponding colour.

### bAb15

U: May I have a table for two people in a cheap price range?  
 S: Any preference for a type of cuisine?  
 U: With British cuisine.  
 S: Where should it be?  
 U: In Rome, thanks.  
 S: Let me look for some options  
 U: Instead could it be with Italian food?  
 S: Sure, is there anything else?  
 U: No, thanks.  
 S: How about Pizza Hut?

---

**SELECT** cuisine, loc, ppl, price, name  
**FROM** restaurant

Table A2: Example of user goal query in SQL formation. The user goal specify a cuisine (i.e., Italian), but in the dialogue the user mention multiple once. To resolve this tie, we select the last mentioned cuisine entities in the dialogue.

- 4. Decide whether the entities in the response are in one of the possible entities in the KB.
- 5. Check whether the entities in the response text answer the given dialogue history or not (you need to make sure that the relation between each entity’s attribute are also correct)
- 6. Count the number of correct entities attributes in the given text and fill the correct\_entity column

### B.3 Human Evaluation Results

In Humanness collected 3 annotations for each sample, while for correctness we used 1 annotation for each sample made by an expert. We take the

### OpenDialKG

U: Any movies similar to Gangs of New York that you can recommend?  
 S: Sure, Quantum of Solace has the same actor Tim Pigott-Smith.  
 U: Is that the one with Daniel Craig?  
 S: Yes, it is a thriller also starred by Daniel Craig.  
 U: I really love thrillers. Any suggestion?  
 S: Daniel Craig also starred in The Girl with the Dragon Tattoo  
 U: Thanks for the suggestion

### MATCH

n1-[ActorsIn]→ n2,  
 n1-[ActorsIn]→ n3,  
 n4-[ActorsIn]→ n3,  
 n4-[ActorsIn]→ n6,  
 n3-[HasGenre]→ n5,  
 n6-[HasGenre]→ n5  
**RETURN** n1, n2, n3,  
 n4, n5, n6

Table A3: Example of user goal query from OpenDialKG (Moon et al., 2019) with CYPHER syntax (Webber, 2012), where the nodes are the requested information in  $R$ , and the labeled edges the constrains in  $C$ .

mean of the annotation score to get the inter-rater agreement score. Our human evaluation reaches statistical significance with 95% confidence interval. We report the human evaluation statistics for each dataset in Table B5. The result of humanness and correctness human evaluation are shown in Figure B2 and Figure B3 respectively.

## C System Comparison

To make a clear distinction of our work to existing task-oriented dialogue systems, we categorize them based on the annotated information and external dependencies used in the pre-processing phase and training-inference phase, such as knowledge base ( $KB$ ), API call for retrieving information( $API$ ),

	Pre-Processing			Training/Testing					Model
	Goal	Span	KB	DST	S-ACT	KB	API	LEX-R	
<i>E2E+Pipelined</i>	✗	✓	✗	✓	✓/✗	✓	✗	✓	Sequicity (Lei et al., 2018), DAMD (Zhang et al., 2019a), Structured Fusion (Mehri et al., 2019), HDSA (Chen et al., 2019), UniConv (Le et al., 2020), Soloist (Peng et al., 2020a), SimpleTOD (Hosseini-Asl et al., 2020), MultiWOZ Benchmark (Budzianowski et al., 2018)
<i>E2E+API+KB</i>	✓	✗	✓	✓	✗	✓	✓	✗	MemoryNet (Bordes and Weston, 2017), Copy-Augmented Seq2Seq (Eric and Manning, 2017), Mem2Seq (Madotto et al., 2018), MLMN (Reddy et al., 2019), GLMP (Wu et al., 2019b), BoSsNet (Raghu et al., 2019), KB-Trs (Haihong et al., 2019)
<i>E2E+GOLD KB</i>	✓	✗	✓	✗	✗	✓	✗	✗	KVRet (Eric et al., 2017a), Mem2Seq (Madotto et al., 2018), KBRet (Qin et al., 2019), Neural Assistant (Neelakantan et al., 2019), GLMP (Wu et al., 2019b), DFF (Qin et al., 2020), GCN (Banerjee and Khapra, 2019),
<i>E2E+KB</i>	✗	✗	✗	✗	✗	✓	✗	✗	Neural Assistant (Neelakantan et al., 2019)
<i>OURS</i>	✓	✓	✓	✗	✗	✗	✗	✗	KE-Dialogue

Table A4: Comparison between different task-oriented methodologies in terms of annotation and mechanism used during pre-processing, training, and inference. *Goal* denotes user goal, *Span* denotes dialogue span, *KB* denotes knowledge base, *DST* denotes dialogue state tracking, *S-ACT* denotes speech act, *API* denotes API call, and *LEX-R* denotes lexicalization for the responses.

Statistics		CamRest	SMD	MWoZ
<b>Humanness</b>	#annotation	3	3	3
	#utterance	150	450	495
	avg. deviation	0.88	0.74	0.85
<b>Correctness</b>	#annotation	1	1	1
	#utterances	147	255	339

Table B5: Human evaluation statistics.

user goal (*Goal*), dialogue span (*Span*), dialogue state tracking (*DST*), speech act (*S-ACT*), and lexicalization response (*LEX-R*). As shown in Table A4, we classify the existing work into four different categories *E2E+Pipelined*, *E2E+API+KB*, *E2E+GOLD KB*, and *E2E+KB*.

Our work is very distinct to all existing works because our approach does not incorporate any annotated information and external dependencies during training and inference time. Our approach utilizes some annotated information only on the pre-processing phase and it trains the model end-to-end with the knowledge-embedded dataset. Our approach is not only removing the dependencies to external dependencies but also eliminate most of the complexity of the whole training-inference process.

## D Experimental Settings

We report our hyper-parameters to train our model in Table D6 for SMD, CAMREST, and OpenDialog and Table D7 for MultiWOZ 2.1.

## E Datasets Information

Table E8 shows the data splits (train/valid/test) and the link to download each dataset.

	GPT2	+KE25	+KE50	+KE75	+KE100
batch size	8	8	8	8	8
grad accu	4	4	4	4	4
lr	6.25e-5	6.25e-5	6.25e-5	6.25e-5	6.25e-5
epoch	30	30	30	30	30
fp16	-	-	-	-	-
max length	150	150	150	150	150
max history	50	50	50	50	50
num layer	12	12	12	12	12
num head	12	12	12	12	12
num emb	768	768	768	768	768
vocab size	50k	50k	50k	50k	50k
params	117M	117M	117M	117M	117M
topk	1	1	1	1	1

Table D6: Hyper-parameters on SMD, CAMREST, and OpenDialog. The experiments were run on several Nvidia 1080Ti.

## F Detailed Experiment Results

We report more detailed results for bAbI-5, SMD, CamRest and MwoZ. Figure F9 shows all detailed results in bAbI dataset. Figure F11 shows all detailed results in SMD dataset. Figure F10 shows all detailed results on CamRest676 dataset. Figure F12 shows all detailed results on MWOZ 2.1 dataset.

## G How many TEMPLATES are enough?

We further analyze our result to see how many TEMPLATES are enough to achieve good performance in the corresponding dataset. In CamRest dataset, as shown in Figure G5, we can see that there is a steep increase from without KE-dialogue to 10 TEMPLATES in term of F1 and a steep improvement from 10 TEMPLATES to 50 TEMPLATES in term of BLEU. This fact sug-



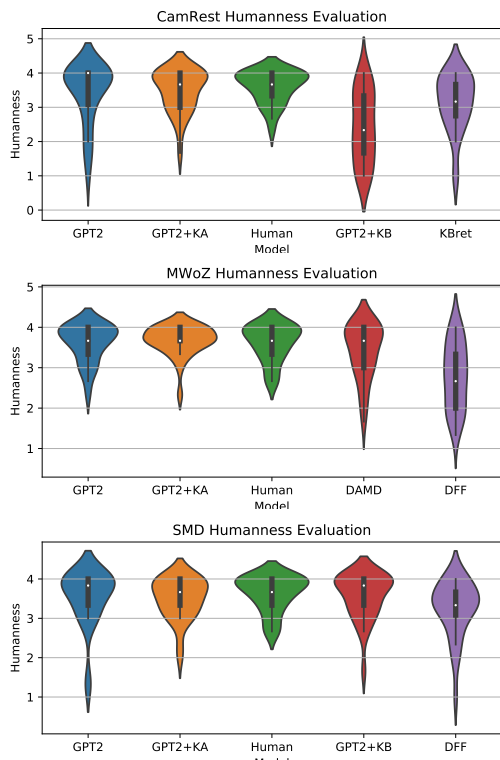


Figure B2: Humanness evaluation in CamRest, MWOZ, and SMD dataset.

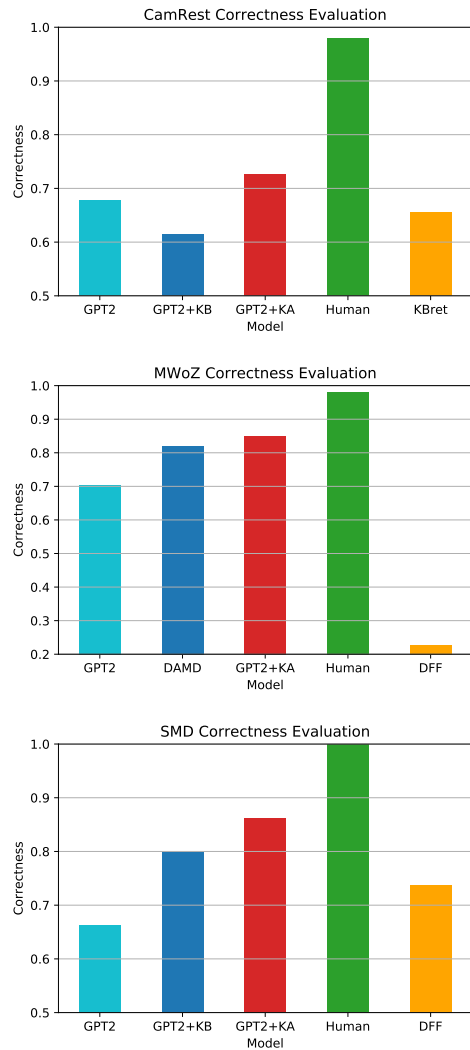


Figure B3: Humanness evaluation in CamRest, MWOZ, and SMD dataset.

gests that 50 TEMPLATES on CamRest dataset is enough to represent the whole dataset. In MWOZ dataset, as shown in Figure G4, with 100 templates the inform and success scores are still increasing while the BLEU score remains stable over TEMPLATES. This suggests that we need more than 100 TEMPLATES to get the optimum benefit from our approach.

In SMD dataset, as shown in G6, in Schedule domain the F1-scores keep increasing steadily until 50 TEMPLATES and slowing down in 75 and 100 TEMPLATES. In Navigation domain there is a steep increase of F1-score from the one without KE-dialogue to the one with 10 TEMPLATES. In weather domain, the F1-score increases steadily from 10 to 100 TEMPLATES. This results suggest on Schedule domain, around 100 TEMPLATE is needed to get the optimal score, while on naviga-

tion domain, only a around 10 to 25 TEMPLATES is required, and Weather domain more than 100 TEMPLATES is required in order to achieve the optimal score.

## H Example of Template Generation

Examples illustrating the KE-DELEX and KE-RELEX process for MultiWOZ, CamRest, and SMD datasets are, respectively, shown in Table H13, Table H14, and Table H15. For CamRest dataset, we remove all of the API calls in the dialogue. For MultiWOZ we keep booking API call and keep the booking reference number as is.

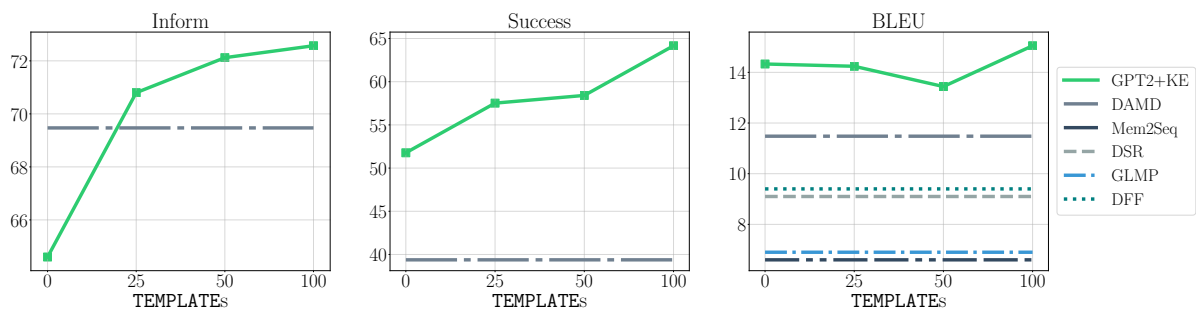


Figure G4: Inform, Success, BLEU score versus number of TEMPLATES in the MultiWOZ dataset.

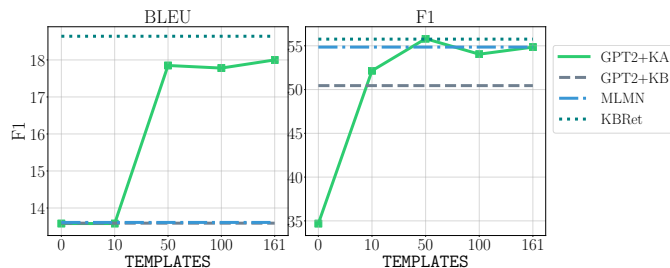


Figure G5: BLEU and F1-Score versus number of TEMPLATES by domain in the CamRest dataset.

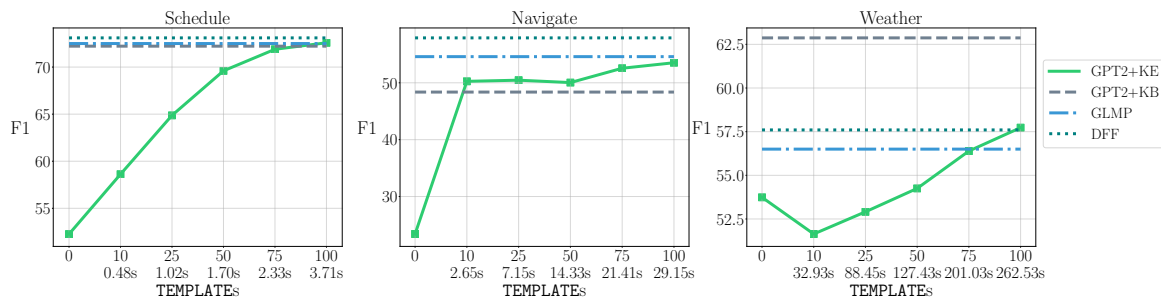


Figure G6: F1-Score versus number of TEMPLATES by domain in the SMD (Eric et al., 2017a) dataset. In the x-axis we report the average fine-tuning time per epoch.

	<b>GPT2</b>	<b>+KE25</b>	<b>+KE50</b>	<b>+KE100</b>
batch size	6	6	6	6
grad accu	3	3	3	3
lr	6.25e-5	6.25e-5	6.25e-5	6.25e-5
epoch	10	10	10	5
fp16	O2	O2	O2	O2
max length	150	150	150	150
max history	50	50	50	50
num layer	12	12	12	12
num head	12	12	12	12
num emb	768	768	768	768
vocab size	50k	50k	50k	50k
params	117M	117M	117M	117M
topk	1	1	1	1

Dataset	Split			Source
	Train	Valid	Test	
bAbI	1,000	1,000	1,000	Website
CAMREST	406	135	135	Github repository
SMD (KVR)	2,425	302	304	Website
MultiWOZ	2,447	204	226	Github repository
attraction single	127	11	12	
hotel single	513	56	67	
restaurant single	1,199	50	62	
taxi single	326	57	52	
train single	282	30	33	
OpenDialog	11,041	1,380	1,380	Facebook Github repository

Table E8: Dataset Statistics and Source.

Table D7: Hyper-parameters on MultiWOZ. The experiments were run on a single Nvidia V100.

Model	Test	Test OOV
QRN <sup>1</sup>	99.60 (-)	67.80 (-)
Mem2Seq <sup>2</sup>	97.90 (69.60)	84.50 (2.30)
BoSsNet <sup>3</sup>	97.30 (65.60)	91.70 (18.50)
GLMP <sup>4</sup>	99.20 (88.50)	92.00 (21.70)
GPT2	90.74 (31.00)	70.14 (0.00)
GPT2+KE 1	93.31 (46.10)	74.75 (2.00)
GPT2+KE 10	99.84 (98.10)	96.84 (77.20)
GPT2+KE 50	99.78 (97.10)	<b>99.60 (95.70)</b>
GPT2+KE 100	<b>99.99 (99.90)</b>	99.01 (94.90)

Table F9: Results on the bAbI dataset.<sup>1</sup> (Seo et al., 2017), <sup>2</sup> (Madotto et al., 2018), <sup>3</sup> (Raghu et al., 2019), <sup>3</sup> (Wu et al., 2019b).

Model	Success	BLEU	F1	Human	Correct
Human	86.08	-	-	3.60	96.97
KB-Trs <sup>1</sup>	-	14.80	45.30	-	-
MLMN <sup>2</sup>	-	13.61	54.85	-	-
BoSsNet <sup>3</sup>	-	15.20	43.10	-	-
KBRet <sup>4</sup>	62.03	<b>18.64</b>	55.76	3.13	77.33
GPT2	30.38	13.58	34.69	3.42	66.67
GPT2+KB	62.03	13.59	50.45	2.42	70.37
GPT2+KE10	62.03	16.55	52.15	-	-
GPT2+KE50	70.89	17.85	<b>55.81</b>	-	-
GPT2+KE100	72.15	17.78	54.04	-	-
GPT2+KE161	<b>74.68</b>	<b>18.00</b>	54.85	<b>3.48</b>	<b>83.50</b>

Table F10: Detailed results on CAMREST dataset. <sup>1</sup>(Haihong et al., 2019). <sup>2</sup>(Reddy et al., 2019). <sup>3</sup>(Raghu et al., 2019). <sup>4</sup>(Qin et al., 2019). We re-evaluate <sup>4</sup> using our script that includes postcode as entity and removes the API-call from the F1-count.

Model	BLEU	Ent.	Nav.	Wea.	Sch.	Hum.	Cor.
KVRet <sup>1</sup>	13.20	48.00	44.50	53.30	62.90	-	-
MLMN <sup>2</sup>	17.10	55.10	41.30	47.00	68.30	-	-
BoSsNet <sup>3</sup>	8.3	35.9	-	-	-	-	-
Mem2Seq <sup>4</sup>	12.20	33.40	20.00	49.30	32.80	-	-
KBRet <sup>5</sup>	13.90	53.70	54.50	52.20	55.60	-	-
KB-Trs <sup>6</sup>	13.90	37.10	23.30	48.20	51.20	-	-
GLMP <sup>7</sup>	13.90	60.70	54.60	56.50	72.50	-	-
DFP <sup>8</sup>	14.40	<b>62.70</b>	<b>57.90</b>	57.60	<b>73.10</b>	3.28	68.90
GPT2	15.60	39.11	23.41	53.74	52.26	<b>3.49</b>	67.05
GPT2+KB	17.03	58.60	48.37	<b>62.87</b>	72.22	3.47	81.03
GPT2+KE 10	14.18	52.88	50.26	51.64	58.62	-	-
GPT2+KE 25	14.22	55.00	50.46	52.91	64.87	-	-
GPT2+KE 50	14.90	56.43	50.04	54.25	69.60	-	-
GPT2+KE 75	16.31	58.79	52.56	56.39	71.89	-	-
GPT2+KE 100	<b>17.35</b>	<b>59.78</b>	<b>53.53</b>	57.73	<b>72.58</b>	3.44	<b>85.56</b>
Human <sup>1</sup>	13.50	60.70	55.20	61.60	64.30	3.54	97.92

Table F11: Results on the SMD (KVR) dataset. <sup>1</sup>Eric et al. (2017b) <sup>2</sup>(Reddy et al., 2019) <sup>3</sup>(Raghu et al., 2019) <sup>4</sup>(Madotto et al., 2018) <sup>5</sup>(Qin et al., 2019) <sup>6</sup>(Haihong et al., 2019) <sup>7</sup>(Wu et al., 2019b) <sup>8</sup>(Qin et al., 2020)

Model	Inform	Success	BLEU	F1	Train	Attraction	Hotel	Rest	Taxi	Human	Correct
Human	-	-	-	-	-	-	-	-	-	3.66	96.85
Mem2Seq <sup>2</sup>	-	-	6.60	21.62	-	22.00	21.00	22.40	-	-	-
DSR <sup>3</sup>	-	-	9.10	-	30.00	28.00	27.00	33.40	-	-	-
GLMP <sup>4</sup>	-	-	6.90	-	32.40	24.40	28.10	38.40	-	-	-
DFP <sup>5</sup>	-	-	9.40	-	35.10	28.10	30.60	<b>40.90</b>	-	2.65	25.53
GPT2	64.60	51.77	14.33	30.38	23.30	15.11	23.56	25.62	89.76	3.51	55.91
GPT2+KE-25	70.80	57.52	14.24	36.96	22.27	43.30	29.74	35.71	87.62	-	-
GPT2+KE-50	72.12	58.41	13.44	37.20	21.95	<b>44.72</b>	30.03	36.10	87.38	-	-
GPT2+KE-100	<b>72.57</b>	<b>64.16</b>	<b>15.05</b>	<b>39.58</b>	<b>23.79</b>	43.32	<b>33.44</b>	37.10	<b>92.38</b>	<b>3.56</b>	<b>73.38</b>
DAMD <sup>1</sup>	85.40	70.40	13.50	-	-	-	-	-	-	-	-
DAMD*	72.12	61.06	11.48	22.58	16.96	31.05	15.50	22.23	55.95	3.31	67.97

Table F12: Detailed results on MultiWOZ dataset. <sup>1</sup>(Zhang et al., 2019a). <sup>2</sup>(Madotto et al., 2018). <sup>3</sup>(Wen et al., 2018). <sup>4</sup>(Wu et al., 2019b). <sup>5</sup>(Qin et al., 2020). \*We evaluate DAMD with our scorer.

TURN	ORIGINAL TEXT	TEMPLATE
USR	i am looking for a particular restaurant . it is called pizza hut city centre .	i am looking for a particular restaurant . it is called <b>[name_0]</b> .
SYS	it s a cheap italian place in the centre at regent street city centre cb21ab . the phone number is 01223323737 . would you like more info ?	it s a <b>[price_0] [food_0]</b> place in the <b>[area_0]</b> at <b>[address_0] [post_0]</b> . the phone number is <b>[phone_0]</b> . would you like more info ?
USR	that sounds like just the place ! can you book me a table for 2 at 1945 on thursday ?	that sounds like just the place ! can you book me a table for 2 at 1945 on thursday ?
SYS-API	restaurant name = 'pizza hut city centre' time = '1945' day = 'thursday' people = '2'	restaurant name = ' <b>[name_0]</b> ' time = '1945' day = 'thursday' people = '2'
API	f3k2pqzz	f3k2pqzz
SYS	you are all set . i have got a table for 2 at 1945 this thursday . your reference number is f3k2pqzz .	you are all set . i have got a table for 2 at 1945 this thursday . your reference number is f3k2pqzz .
USR	ok sounds good . thank you !	ok sounds good . thank you !
SYS	you are welcome . is there anything else i can assist you with today ?	you are welcome . is there anything else i can assist you with today ?
USR	that is all . thank you for your help !	that is all . thank you for your help !
SYS	you are welcome . have a great day .	you are welcome . have a great day .

USER GOAL QUERY

**SELECT** area, food, price, name, address, phone, postcode **FROM** Restaurant

Area	Food	Price	Name	Address	Phone	Postcode
centre	italian	cheap	pizza hut city centre	Regent Street City Centre	01223323737	cb21ab
centre	indian	cheap	kohinoor	74 Mill Road City Centre	01223323639	cb12as
centre	chinese	expensive	hk fusion	21 Burleigh Street City Centre	01223355909	cb11dg
south	modern european	moderate	restaurant alimentum	152 - 154 Hills Road	01223413000	cb28pb

TURN	RELEX 1	RELEX 2
USR	i am looking for a particular restaurant . it is called <b>kohinoor</b> .	i am looking for a particular restaurant . it is called <b>restaurant alimentum</b> .
SYS	it s a <b>cheap indian</b> place in the <b>centre</b> at <b>74 mill road city centre cb12as</b> . the phone number is <b>01223323639</b> . would you like more info ?	it s a <b>moderate modern european</b> place in the <b>south</b> at <b>152 - 154 hills road cb28pb</b> . the phone number is <b>01223413000</b> . would you like more info ?
USR	that sounds like just the place ! can you book me a table for 2 at 1945 on thursday ?	that sounds like just the place ! can you book me a table for 2 at 1945 on thursday ?
SYS-API	restaurant name = ' <b>kohinoor</b> ' time = '1945' day = 'thursday' people = '2'	restaurant name = ' <b>restaurant alimentum</b> ' time = '1945' day = 'thursday' people = '2'
API	f3k2pqzz	f3k2pqzz
SYS	you are all set . i have got a table for 2 at 1945 this thursday . your reference number is f3k2pqzz .	you are all set . i have got a table for 2 at 1945 this thursday . your reference number is f3k2pqzz .
USR	ok sounds good . thank you !	ok sounds good . thank you !
SYS	you are welcome . is there anything else i can assist you with today ?	you are welcome . is there anything else i can assist you with today ?
USR	that is all . thank you for your help !	that is all . thank you for your help !
SYS	you are welcome . have a great day .	you are welcome . have a great day .

Table H13: MultiWOZ 2.1 KE-DELEX and KE-RELEX Examples

TURN	ORIGINAL TEXT	TEMPLATE
USR	i want a moderately priced restaurant in the east part of town .	i want a <b>[price.1]</b> ly priced <b>[type.1]</b> in the <b>[area.1]</b> part of town .
SYS	would you prefer indian food or italian food ?	would you prefer <b>[food.2]</b> food or <b>[food.3]</b> food ?
USR	i do n't care . give me their phone number .	i do n't care . give me their phone number .
SYS	curry prince is a moderately priced restaurant in the east part of town that serves indian food . would you like the phone number ?	<b>[name.2]</b> is a <b>[price.2]</b> ly priced <b>[type.2]</b> in the <b>[area.2]</b> part of town that serves <b>[food.2]</b> food . would you like the phone number ?
USR	yes give me the phone number .	yes give me the phone number .
SYS	the phone number for Curry Prince is 01223566388 .	the phone number for <b>[name.2]</b> is <b>[phone.2]</b> .
USR	awesome , thank you ! goodbye !	awesome , thank you ! goodbye !
SYS	you 're welcome , goodbye .	you 're welcome , goodbye .

USER GOAL QUERY

**SELECT** area, food, price, name, phone **FROM** Restaurant

Area	Food	Price	Name	Address	Phone	Postcode	Location
east	indian	moderate	curry prince	451 newmarket road	01223566388	cb58jj	52.213072,0.149771
east	italian	moderate	pizza hut	cambridge retail park	01223323737	cb58wr	52.209742,0.146975
east	indian	moderate	rajmahal	7 barnwell road	01223244955	cb58rg	52.211144,0.16324

TURN	RELEX 1	RELEX 2
USR	i want a <b>moderately</b> priced <b>restaurant</b> in the <b>east</b> part of town .	i want a <b>moderately</b> priced <b>restaurant</b> in the <b>east</b> part of town .
SYS	would you prefer <b>indian</b> food or <b>italian</b> food ?	would you prefer <b>italian</b> food or <b>indian</b> food ?
USR	i do n't care . give me their phone number .	i do n't care . give me their phone number .
SYS	<b>rajmahal</b> is a <b>moderately</b> priced <b>restaurant</b> in the <b>east</b> part of town that serves <b>indian</b> food . would you like the phone number ?	<b>pizza hut</b> is a <b>moderately</b> priced <b>restaurant</b> in the <b>east</b> part of town that serves <b>italian</b> food . would you like the phone number ?
USR	yes give me the phone number .	yes give me the phone number .
SYS	the phone number for <b>rajmahal</b> is <b>01223244955</b> .	the phone number for <b>pizza hut</b> is <b>01223323737</b> .
USR	awesome , thank you ! goodbye !	awesome , thank you ! goodbye !
SYS	you 're welcome , goodbye .	you 're welcome , goodbye .

Table H14: CamRest Delex and Relex Examples

TURN	ORIGINAL TEXT	TEMPLATE
USR	where is the closest rest stop ?	where is the closest <b>[poi_type_1]</b> ?
SYS	the westin is 2 miles away	<b>[poi_11]</b> is <b>[distance_11]</b> away
USR	what is the address ?	what is the address ?
SYS	the westin is located at 329 el camino real	<b>[poi_11]</b> is located at <b>[poi_address_11]</b> .
USR	thanks set the navigation to there please	thanks set the navigation to there please
SYS	you are welcome , i set the navigation	you are welcome , i set the navigation

User Goal Query

**SELECT** type, poi, dist, addr **FROM** navigation **GROUP BY** type **HAVING** dist = **MIN**(dist)

POI	POI Type	Address	Distance	Traffic
whole foods	grocery store	819 alma st	2 miles	heavy traffic
hotel keen	rest stop	578 arbol dr	3 miles	no traffic
home	home	338 alester ave	3 miles	no traffic
round table	pizza restaurant	113 anton ct	4 miles	heavy traffic
mandarin roots	chinese restaurant	271 springer street	3 miles	heavy traffic

TURN	RELEX 1	RELEX 2
USR	where is the closest <b>grocery store</b> ?	where is the closest <b>home</b> ?
SYS	<b>whole foods</b> is <b>2 miles</b> away	<b>home</b> is <b>3 miles</b> away
USR	what is the address ?	what is the address ?
SYS	<b>whole foods</b> is located at <b>819 alma st</b>	<b>home</b> is located at <b>338 alester ave</b>
USR	thanks set the navigation to there please	thanks set the navigation to there please
SYS	you are welcome , i set the navigation	you are welcome , i set the navigation

Table H15: SMD Delex and Relex Example