Contextualize Knowledge Bases with Transformer for End-to-end Task-Oriented Dialogue Systems

Yanjie Gou[1], **Yinjie Lei**[1]*, **Lingqiao Liu**[2], **Yong Dai**[3], **Chunxu Shen**[4]
[1]College of Electronics and Information Engineering, Sichuan University, China
[2]School of Computer Science, The University of Adelaide, Australia
[3]University of Electronic Science and Technology of China, China [4]Tencent
gouyanjie@stu.scu.edu.cn, yinjie@scu.edu.cn, lingqiao.liu@adelaide.edu.au
daiyongya@yahoo.com, lineshen@tencent.com

Abstract

Incorporating knowledge bases (KB) into end-to-end task-oriented dialogue systems is challenging, since it requires to properly represent the entity of KB, which is associated with *its KB context* and *dialogue context*. The existing works represent the entity with only perceiving ***a part of*** its KB context, which can lead to the less effective representation due to the information loss, and adversely favor KB reasoning and response generation. To tackle this issue, we explore to ***fully contextualize*** the entity representation by dynamically perceiving *all the relevant entities* and *dialogue history*. To achieve this, we propose a COntext-aware Memory Enhanced Transformer framework (COMET), which treats the KB as a sequence and leverages a novel ***Memory Mask*** to enforce the entity to only focus on its relevant entities and dialogue history, while avoiding the distraction from the irrelevant entities. Through extensive experiments, we show that our COMET framework can achieve superior performance over the state of the arts.

1 Introduction

Task-oriented dialogue systems aim to achieve specific goals such as hotel booking and restaurant reservation. The traditional pipelines (Young et al., 2013; Wen et al., 2017) consist of natural language understanding, dialogue management, and natural language generation modules. However, designing these modules often requires additional annotations such as dialogue states. To simplify this procedure, the end-to-end dialogue systems (Eric and Manning, 2017) are proposed to incorporate the KB (normally relational databases) into the learning framework, where the KB and dialogue history can be directly modeled for response generation, without the explicit dialogue state or dialogue action.

Poi	Poi type	Traffic	Address	Distance
Stanford Express Care	hospital	moderate	214 El Camino Real	2 miles
Tom's house	friend's house	no	580 Van Ness Ave	6 miles
Philz	coffee or tea place	no	583 Alester Ave	4 miles
5672 Barringer Street	certain address	no	5672 Barringer Street	2 miles

User	Where does my friend live ?
System	Tom's house is 6 miles away at 580 Van Ness Ave .
User	Is that the fastest route ?
System	I'll send the route with no traffic on your screen , drive carefully !

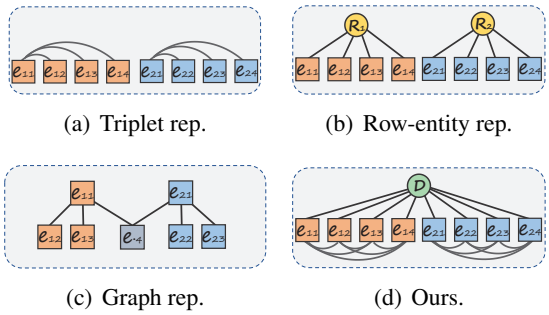
Table 1: An example in SMD dataset (Eric et al., 2017). The top is the entities in KB and the bottom is a two-turn dialogue between the user and system.

An example of the end-to-end dialogue systems is shown in Tab. 1. When generating the second response about the "*traffic info*": (1) the targeted entity "*no traffic*" is associated with its same-row entities (KB context) like "*Tome's house*", "*friend's house*" and "*6 miles*". These entities can help with enriching the information of its representation and modeling the structure of KB. (2) Also, the entity is related to the dialogue history (dialogue context), which provides clues about the goal-related row (like "*Tom's house*" and "*580 Van Ness Ave*" in the first response). These clues can be leveraged to further enhance the corresponding representations and activate the targeted row, which benefits the retrieval of "*no traffic*". Therefore, how to ***fully contextualize*** the entity with its KB and dialogue contexts, is the key point of end-to-end dialogue systems (Madotto et al., 2018; Wu et al., 2019; Qin et al., 2020), where the full-context enhanced entity representation can make the reasoning over KB and the response generation much easier.

However, the existing works can only contextualize the entity with perceiving parts of its KB context and ignoring the dialogue context: (1) (Madotto et al., 2018; Wu et al., 2019; Qin et al., 2020) rep-

*Corresponding author

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 4300–4310
November 7–11, 2021.



(a) Triplet rep. (b) Row-entity rep. (c) Graph rep. (d) Ours.

Figure 1: Four ways to represent the KB, where $e_{i,j}$ means the entity representation for the j-th entity of the i-th row; R_i means the row representation of the i-th row; $e_{.,j}$ means the entities shared between different row, like "*no traffic*" in Tab. 1; D means the dialogue context. *Note that the existing three representations (a-c) only consider parts of the KB context and ignore the dialogue context, whereas our method (d) can* ***fully contextualize*** *the entity with both of them.*

resent an entity as a triplet (cf. Fig. 1(a)), i.e., (Subject, Relation, Object). However, breaking one row into several triplets can only model the relation between two entities, whereas the information from other same-row entities and dialogue history are ignored. (2) (Gangi Reddy et al., 2019; Qin et al., 2019) represent KB in a hierarchical way, i.e., the row and entity-level representation (cf. Fig. 1(b)). This representation can only partially eliminate this issue at the row level. However, at the entity level, the entity can only perceive the information of itself, which is isolated with other KB and dialogue contexts. (3) (Yang et al., 2020) converts KB to a graph (cf. Fig. 1(c)). However, they fails to answer what is the optimal graph structure for KB. That indicates their graph structure may need manual design[1]. Also, the dialogue context is not encoded into the entity representation, which can also lead to the suboptimal entity representation. To sum up, these existing methods ***can not*** fully contextualize the entity, which leads to vulnerable KB reasoning and response generation.

In this work, we propose COntext-aware Memory Enhanced Transformer (**COMET**), *which provides a unified solution to* ***fully contextualize*** *the entity with the awareness of both the KB and dialogue contexts* (shown in Fig. 1(d)). The key idea of COMET is that: a **Memory-Masked** Encoder is used to encode the entity sequence of KB, along with the information of dialogue history. The designed Memory Mask is utilized to ensure the entity can only interact with its same-row entities and the information in dialogue history, whereas the distractions from other rows are prohibited.

More specifically, (1) *for the KB context*, we represent the entities in the same row as a sequence. Then, a Transformer Encoder (Vaswani et al., 2017) is leveraged to encode them, where the same-row entities can interact with each other. Furthermore, to retain the structure of KB and avoid the distractions from the entities in different rows, we design a ***Memory Mask*** (shown in Fig. 3) and incorporate it into the encoder, which only allows the interactions between the same-row entities. (2) *For the dialogue context*, we create a Summary Representation (*Sum. Rep*) to summarize the dialogue history, which is input into the encoder to interact with the entity representations (gray block in Fig. 2). We also utilize the Memory Mask to make the *Sum. Rep* overlook all of the entities for better entity representations, which will serve as the context-aware memory for further response generation.

By doing so, we essentially extend the entity of KB to $(\mathcal{N}+1)$-tuple representation, where $\mathcal{N}$ is the number of entities in one row and "1" is for the Sum. Rep of the dialogue history. By leveraging the KB and dialogue contexts, our method can effectively model the information existing in KB and activate the goal-related entities, which benefits the entity retrieval and response generation. Please note that the function of fully contextualizing entity is unified by the designed Memory Mask scheme, which is the key of our work.

We conduct extensive experiments on two public benchmarks, i.e., SMD (Eric et al., 2017; Madotto et al., 2018) and Multi-WOZ 2.1 (Budzianowski et al., 2018; Yang et al., 2020). The experimental results demonstrate significant performance gains over the state of the arts. It validates that contextualizing KB with Transformer benefits entity retrieval and response generation.

In summary, our contributions are as follows:

- To the best of our knowledge, we are the first to fully contextualize the entity representation with both the KB and dialogue contexts, for end-to-end task-oriented dialogue systems.
- We propose Context-aware Memory Enhanced Transformer, which incorporates a designed Memory Mask to represent entity with

[1]For instance, on the SMD dataset, they only activate the edges between the primary key ("*poi*") and other keys(e.g., "*address*") in the Navigation domain, but assign a fully-connected graph to the Schedule domain.

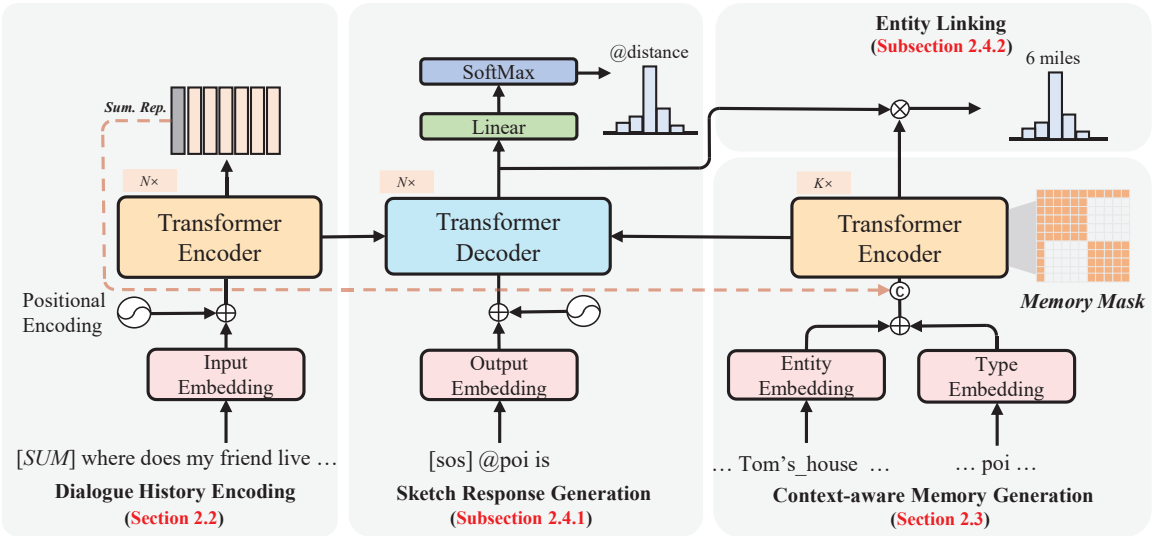


Figure 2: Overview of **COMET**. The gray block in the top left means Sum. Rep of dialogue history, *which is used as the input for the Memory Generation.* © means concatenation. The detailed construction of the Memory Mask can be found in Fig. 3.

awareness of both the relevant entities and dialogue history.

- Extensive experiments demonstrate that our method gives a state-of-the-art performance.

2 Methodology

In this section, we first introduce the general workflow for this task. Then, we elaborate on each part of COMET, i.e., the Dialogue History Encoder, Context-aware Memory Generation, and Response Generation Decoder (as depicted in Fig. 2). Finally, the objective function will be introduced.

2.1 General Workflow

Given a dialogue history with k turns, which is denoted as $\mathcal{H} = \{u_1, s_1, u_2, s_2, ..., u_k\}$ (u_i and s_i denote the i-th turn utterances between the user and the system), the goal of dialogue systems is to generate the k-th system response s_k with an external KB $\mathcal{B} = \{[b_{11}, ..., b_{1c}], ..., [b_{r1}, ..., b_{rc}]\}$, which has r rows and c columns. Formally, the procedure mentioned above is defined as:

$$p(s_k|\mathcal{H}, \mathcal{B}) = \prod_{i=1}^{n} p(s_{k,t}|s_{k,1}, ..., s_{k,t-1}, \mathcal{H}, \mathcal{B}),$$

where we first derive the dialogue history representation (Section 2.2) and generate the Context-aware Memory, a.k.a., contextualized entity representation (Section 2.3), where these two parts will be used to generate the response s_k (Section 2.4).

2.2 Dialogue History Encoder

We first transform $\mathcal{H}$ into the word-by-word form with a special token $[SUM]$: $\hat{\mathcal{H}} = \{x_1, x_2, ..., x_n\}$, $x_1 = [SUM]$, which is used to globally aggregate information from $\mathcal{H}$.

Then, the sequence $\hat{\mathcal{H}}$ is encoded by a standard Transformer Encoder and generate the dialogue history representation H_N^{enc}, where $H_{N,1}^{enc}$ is denoted as the Summary Representation (*Sum. Rep*) of the dialogue history.[2] *It will be used to make the memory aware of the dialogue context.*

2.3 Context-aware Memory Generation

In this subsection, we describe how to "***fully contextualize KB***". That is, the *Memory Mask* is leveraged to ensure the entities of KB with the awareness of all of its related entities and dialogue history, which is the key contribution of our method.

2.3.1 Memory Generation

Different from existing works which fail to contextualize all the useful context information for the entity representation, we treat KB as a sequence, along with *Sum. Rep*. Then, a Transformer Encoder with the Memory Mask is utilized to model it, which can dynamically generate the entity representation with the awareness of its all favorable contexts, i.e., the same-row entities and dialogue history, while blocking the distraction from the

[2]This module is as same as the standard Transformer Encoder, please refer to (Vaswani et al., 2017) for more details.

irrelevant entities. The procedure of memory generation is as follows.

Firstly, the entities in the KB $\mathcal{B}$ is flatten as a memory sequence, i.e., $\mathcal{M} = [b_{11}, ..., b_{1c}, ..., b_{r1}, ..., b_{rc}] = [m_1, m_2, ..., m_{|\mathcal{M}|}]$, where the memory entity m_i means an entity of KB in the k-th row. By doing so, the Memory-Masked Transformer Encoder can interact the same-row entities with each other while retaining the structure information of KB.[3]

Then, $\mathcal{M}$ will be transformed into the entity embeddings, i.e., $E = [e^m_1, ..., e^m_{|\mathcal{M}|}]$, where e^m_i corresponds to m_i in $\mathcal{M}$ and it is the sum of the word embedding u_i and the type embedding t_i, i.e., $e^m_i = u_i + t_i$. Note that, the entity types are the corresponding column names, e.g., "*poi_type*" in Table 1. For the entities which have more than one token, we simply treat them as one word, e.g., "*Stanford Exp*" $\rightarrow$ "*Stanford_Exp*".

Next, the entity embeddings are concatenated with the *Sum. Rep* from the Dialogue History Encoder, i.e. $E_0 = [H^{enc}_{N,1}; E]$. The purpose of introducing $H^{enc}_{N,1}$ is that it passes the information from the dialogue history and further enhances the entity representation with the dialogue context.

Finally, E_0 and the Memory Mask M^{mem} are used as the input of the Transformer Encoder ($tf\_enc(\cdot)$) to generate the context-aware memory (a.k.a, contextualized entity representation):

$$E_l = tf\_enc(E_{l-1}, M^{mem}), l \in [1, K],$$

where K is the total number of Transformer Encoder layers. $E_K \in \mathbb{R}^{(|\mathcal{M}|+1)\times d_m}$ is the generated memory, which is queried when generating the response for entity retrieval.

2.3.2 Memory Mask Construction

To highlight, we design a special Memory Mask scheme to take ALL the contexts grounded by the entity into account, where the Memory Mask ensures that the entity can only attend to its context part, which is the key contribution of this work. This is in contrast to the standard Transformer Encoder, where each entity can attend to all of the other entities. The rationale of our design is that by doing so, we can avoid the noisy distraction of the non-context part.

Formally, $M^{mem} \in \mathbb{R}^{(|\mathcal{M}|+1)\times(|\mathcal{M}|+1)}$ is defined as:

$$M^{mem}_{i,j} = \begin{cases} 1, & if\ \mathcal{M}_{i-1}, \mathcal{M}_{j-1} \in b_k, \\ 1, & if\ i\ or\ j = 1, \\ -\infty, & else. \end{cases}$$

A detailed illustration of the Memory Mask construction is shown in Fig. 3. With this designed Memory Mask, a masked attention mechanism is leveraged to make the entity only attend the entities within the same row and the *Sum. Rep*.

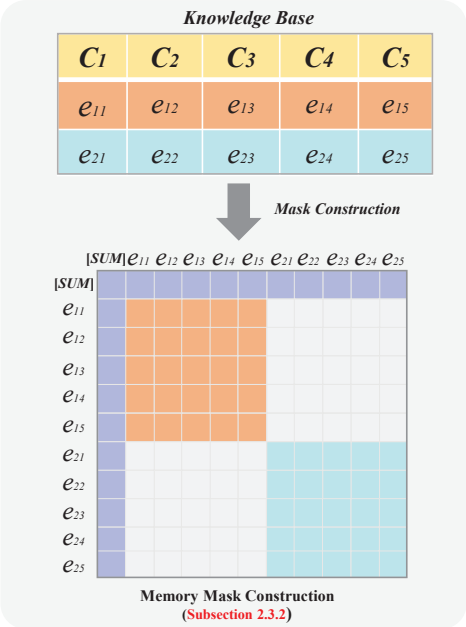


Figure 3: The Construction of Memory Mask. C_i means the column name (e.g., "*Poi*"). e_{ij} means the j-th entity of i-th row. [*SUM*] means the Sum. Rep. Only two rows of KB are shown for simplicity.

2.4 Response Generation Decoder

Given the dialogue history representation H^{enc}_N and generated memory E_K, the decoder will use them to generate the response for a specific query. In COMET, we use a modified Transformer Decoder, which has two cross attention modules to model the information in H^{enc}_N and E_K, respectively. Then, a gate mechanism is leveraged to adaptively fuse H^{enc}_N and E_K for the decoder, where the response generation is tightly anchored by them.

Following (Wu et al., 2019; Qin et al., 2020; Yang et al., 2020), we first generate a sketch response that replaces the exact slot values with sketch tags.[4] Then, the decoder links the entities in

[3] When the memory sequence is long, some existing methods like the linear attention (Kitaev et al., 2020) can be used to tackle the issue of $\mathcal{O}(N^2)$ complexity of Self Attention.

[4] For instance, "Tom's house is 6 miles away at 580 Van Ness Ave ." $\rightarrow$ "@poi is @distance away at @address.".

the memory to their corresponding slots.

2.4.1 Sketch Response Generation

For the k-th turn generating sketch response $\mathcal{Y} = [y_1, ...y_{t-1}]$, it is converted to the word representation $H_0^{dec} = [w_1^d, ..., w_{t-1}^d]$. $w_i^d = v_i + p_i$, where v_i and p_i means the word embedding and absolute position embedding of i-th token in $\mathcal{Y}$.

Afterward, N-stacked decoder layers are applied to decode the next token with the inputs of H_0^{dec}, E_K and H_N^{enc}. The process in one decoder layer can be expressed as:

$$
\begin{aligned}
H_l^{d-d} &= MHA(H_{l-1}^{dec}, H_{l-1}^{dec}, H_{l-1}^{dec}, M^{dec}), \\
H_l^{d-e} &= MHA(H_l^{d-d}, H_N^{enc}, H_N^{enc}), \\
H_l^{d-m} &= MHA(H_l^{d-d}, E_K, E_K), \\
g &= sigmoid(FC(H_l^{d-m})), \\
H_l^{agg} &= g \odot H_l^{d-e} + (1-g) \odot H_l^{d-m}, \\
H_l^{dec} &= FFN(H_l^{agg}),\ l \in [1, N],
\end{aligned}
$$

where the input $\{Q, K, V, M\}$ of the Multi-Head Attention $MHA(Q, K, V, M)$ means the query, key, value, and optional attention mask. $FFN(\cdot)$ means the Feed-Forward Networks. M^{dec} is the decoder mask, so as to make the decoded word can only attend to the previous words. $FC(\cdot)$ is a fully-connected layer to generate the gating signals, which maps a d_m-dimension feature to a scalar. N is the number of the total decoder layers.

After obtaining the final H_N^{dec}, the posterior distribution for the t-th token, $p_t^v \in \mathbb{R}^{|V|}$ ($|V|$ denotes the vocabulary size), is calculated by:

$$
p_t^v = softmax(H_{N,t-1}^{dec} W_v + b_v).
$$

2.4.2 Entity Linking

After the sketch response generation, we replace the sketch tags with the entities in the context-aware memory. We denote the representation from the decoder at the t-th time step, i.e., the t-th token, as $H_{N,t}^{dec}$, and represent the time steps that need to replace sketch tags with entities as $\mathcal{T}$. The probability distribution over all possible linked entities can then be calculated by

$$
p_t^s = softmax(H_{N,t}^{dec} E_K^T), \quad \forall t \in \mathcal{T}
$$

where E_K means the final generated memory.

2.5 Objective Function

For the training process of COMET, we use the the cross-entropy loss to supervise the response generation and entity linking[5].

Moreover, we propose an additional regularization term to further regularize p_t^s. The regularization is based on the prior knowledge that for a given response, only a small subset of entities should be linked. Formally, we construct the following entity linking probability matrix $\mathbf{P}^s = [p_{t_1}^s, p_{t_2}^s, ..., p_{t_{|\mathcal{T}|}}^s]$ and minimize its $L_{2,1}$-norm (Nie et al., 2010):

$$
L_r = \sum_{i=1}^{|\mathcal{M}|} \sqrt{\sum_{t \in \mathcal{T}} (p_{t,i}^s)^2},
$$

where $p_{t,i}^s$ denotes the i-th dimension of p_t^s. This regularization term can encourage the network to select a small subset of entities to generate the response. The same idea has been investigated in (Nie et al., 2010) for multi-class feature selection.

Finally, COMET is trained by jointly minimizing the combination of the above three losses.

3 Experiments

3.1 Datasets

Two public multi-turn task-oriented dialogue datasets are used to evaluate our model, i.e., SMD[6] (Eric et al., 2017) and Multi-WOZ 2.1[7] (Budzianowski et al., 2018). *Note that, for Multi-WOZ 2.1, to accommodate end-to-end settings, we use the revised version released by (Yang et al., 2020), which equips the corresponding KB to every dialogue.* We follow the same partition as (Madotto et al., 2018) on SMD and (Yang et al., 2020) on Multi-WOZ 2.1.

3.2 Experimental Settings

The dimension of embeddings and hidden vectors are all set to 512. The number of layers (N) in Dialogue History Encoder and Response Generation Decoder is set to 6. The number of layers for Context-aware Memory Generation (K) is set to 3. The number of heads in each part of COMET is set to 8. A greedy strategy is used without beam-search during decoding. The Adam optimizer (Kingma

[5] The label construction procedure of the entity linking module can be found in Appendix A.1.

[6] https://github.com/jasonwu0731/GLMP/tree/master/data/KVR

[7] https://github.com/shiquanyang/GraphDialog/tree/master/data/MULTIWOZ2.1

	SMD					Multi-WOZ2.1					
Model	BLEU	F1	F1-Sch.	F1-Wea.	F1-Nav.	BLEU	F1	F1-Res.	F1-Att.	F1-Hot.	F1-Tra.
Mem2Seq	12.6	33.4	49.3	32.8	20.0	4.1	3.2	2.9	2.1	4.5	1.5
KB-Transformer	13.9	37.1	51.2	48.2	23.3	-	-	-	-	-	-
KB-Retriever	13.9	53.7	55.6	52.2	54.5	-	-	-	-	-	-
GLMP	13.9	60.7	72.9	56.5	54.6	4.3	6.7	11.4	9.4	3.9	3.5
DF-Net	14.4	62.7	73.1	57.6	**57.9**	-	-	-	-	-	-
GraphDialog	13.7	60.7	72.8	55.2	54.2	6.2	11.3	16.0	14.1	10.8	4.4
COMET (*Ours*)	**17.3**	**63.6**	**77.6**	**58.3**	56.0	**8.3**	**18.6**	**27.5**	**17.9**	**15.2**	**9.8**

Table 2: BLEU and Entity F1 comparison of COMET with other counterparts. The best results are in **bold font** and the second-best results are underlined. The results on the SMD and Multi-WOZ 2.1 datasets are adopted from (Qin et al., 2020) and (Yang et al., 2020), respectively.

and Ba, 2014) is used to train our model from scratch with a learning rate of $1e^{-4}$. More details about the hyper-parameter settings can be found in Appendix A.2.

3.3 Baselines

We compare COMET with the following methods:

- **Mem2Seq** (*Triplet*) (Madotto et al., 2018): Mem2Seq incorporates the multi-hop attention mechanism in memory networks into the pointer networks.
- **KB-Transformer** (*Triplet*) (E. et al., 2019): KB-Transformer combines a Multi-Head Key-Value memory network with Transformer.
- **KB-Retriever** (*Row-entity*) (Qin et al., 2019): KB-retriever improves the entity-consistency by first selecting the target row and then picking the relevant column in this row.
- **GLMP** (*Triplet*) (Wu et al., 2019): GLMP uses a global memory encoder and a local memory decoder to incorporate the external knowledge into the learning framework.
- **DF-Net** (*Triplet*) (Qin et al., 2020): DF-Net applies a dynamic fusion mechanism to transfer knowledge in different domains.
- **GraphDialog** (*Graph*) (Yang et al., 2020): GraphDialog exploits the graph structural information in KB and in the dependency parsing tree of the dialogue.

3.4 Results

Following the existing works (Qin et al., 2020; Yang et al., 2020), we use the *BLEU* and *Entity F1* metrics to evaluate model performance. The results are shown in Tab. 2.

It is observed that: COMET achieves the best performance over both datasets, which indicates that our COMET framework can better leverage the information in the dialogue history and external KB, to generate more fluent responses with more accurate linked entities. Specifically, for the *BLEU* score, it outperforms the previous methods by 2.9% on the SMD dataset and 2.1% on the Multi-WOZ 2.1 dataset, at least. Also, COMET achieves the highest *Entity F1* score on both datasets. That is, the improvements of 0.9% and 7.3% are attained on the SMD and Multi-WOZ 2.1 datasets, respectively. In each domain of the two datasets, improvement or competitive performance can be clearly observed. The results indicate the superior of our COMET framework.

To highlight, KB-Transformer (E. et al., 2019) also leverages Transformer, but our COMET outperforms it by a large margin. On the SMD dataset, the *BLEU* score of COMET is higher than that of KB-Transformer by 3.4%. The improvement introduced by COMET on *Entity F1* score is as significant as 26.5%. This shows naively introducing Transformer to the end-to-end dialogue system will not necessarily lead to higher performance. A careful design of the whole dialogue system, such as our proposed one, plays a vital role.

3.5 Ablation Study

In this subsection, we first investigate the effects of the different components, i.e., the Memory Mask, Sum. Rep, gate mechanism, and $L_{2,1}$-norm regularization (Tab. 3). Then, we design careful experiments to further demonstrate the effect of the Memory Mask, which is the key contribution of this work: (1) we replace the context-aware memory of COMET with the existing three representations of KB, (i.e., triplet, row-entity, and graph) to show the superior of the fully contextualized entity (Tab. 4). (2) We also replace our Memory Mask with the full attention layer by layer, which further shows

the importance of our Memory Mask (Tab. 5). Our ablation studies are based on the SMD dataset.

Model	BLEU	Entity F1	Δ
COMET	17.3	63.6	-
w/o Memory Mask	15.4	49.6	14.0
w/o Sum. Rep	17.0	61.4	2.2
only use H_N^{enc} (gate)	17.2	61.1	2.5
only use E_K (gate)	17.1	61.4	2.2
w/o $L_{2,1}$-norm	17.4	62.3	1.3

Table 3: The effects of different components.

The effects of the key components in the COMET framework are reported in Tab. 3. As observed, removing any key component of the COMET, both the *BLEU* and *Entity F1* metrics degrade to some extend. More specifically: (1) If the Memory Mask is removed, the *Entity F1* score drops to 49.6. This significant discrepancy demonstrates the importance of restricting self-attention as our designed Memory Mask did. (2) For the variant without the Sum. Rep, the *Entity F1* score drops to 61.4. That indicates the effectiveness of contextualizing the KB with the dialogue history, which can further boost the performance. (3) We also remove the gate and only use the information from the dialogue history (H_N^{enc}) or memory (E_K). We can see that the former case can only achieve 61.1 while the latter case achieves 61.4 of the *Entity F1* score. It is obvious that using the gate mechanism to fuse both information sources is helpful for the entity linking. (4) When removing the $L_{2,1}$-norm, the performance also drops to 62.3, which means regularizing the entity-linking distribution can further benefit the performance.

Model	BLEU	F1	F1-Sch.	F1-Wea.	F1-Nav.
Context-aware memory	**17.3**	**63.6**	**77.6**	**58.3**	**56.0**
Only KB context	17.0	61.4	75.5	55.2	54.4
Triplet	14.9	59.8	73.1	54.0	53.0
Row&Ent	13.0	41.4	51.2	54.6	19.3
Graph	14.4	56.7	71.6	48.7	50.4

Table 4: The performance of replacing the context-aware memory with Triplet, Row-Ent and Graph representations in COMET. Note that in the second row, we also report the result of a variant which only considers the KB context and ignores the dialogue context.

We also replace our context-aware memory with other ways of representing KB, while other parts of our framework keep unchanged[8]. The result is reported in Tab. 4. It is observed that, After replacing our context-aware memory with the existing three representations of KB, the performance drops a lot in all the metrics, where the *BLEU* score drops 2.4% and the *Entity F1* score drops 3.8% at least. Besides, the result of the variant which only considers the KB context part (i.e., w/o Sum. Rep), is also reported, so as to further fairly compare with the aforementioned KB representations. The result shows that only considering the KB context, our method can still outperform other KB representations by 1.6% of *Entity F1* at least. That further indicates the fully contextualizing entity with its relevant entity and the dialogue history, can better represent the KB for dialogue systems.

Scheme	BLEU	Entity F1	Δ
MMM	17.3	63.6	-
MMF	16.5	61.2	2.4
MFF	16.5	59.1	4.5
FFF	15.4	49.6	14.0

Table 5: The performance of replacing the Memory Mask with the full attention. The meanings of the scheme names are that the Memory Mask (**M**) is replaced with the Full attention (**F**).

We also conduct the experiment which replaces the Memory Mask with the full attention, layer by layer. That is, the first (n-k) layers use the proposed Memory Mask (**M**) and the last k layers use the full attention (**F**). As shown in Tab. 5, the more full attention is added, the more performance of COMET drops in all of the metrics since the full attention introduces too much distraction from other rows. The result further indicates that the Memory Mask is indeed a better choice which takes the inductive bias of KB into account.

Note that we also explore other Memory Mask schemes, but these schemes can not further boost the performance, where the results are omitted due to the page limitation. For further improvement, more advanced techniques like Pre-trained Model (Devlin et al., 2018; Radford et al., 2019) may be needed to deeply understand the dialogue and KB context, which we leave for future work.

3.6 Case Study

To demonstrate the superiority of our method, several examples on the SMD test set, which are generated by our COMET and the existing state of the arts GLMP (Wu et al., 2019) and DF-Net (Qin et al., 2020), are given in Tab. 6. As reported, compared with GLMP and DF-Net, COMET can generate

[8]The implementation details are in Appendix A.3.

Query&Response Example
Goal: [yoga_activity, 11am, thursday, alex]
Query: what time do i go to yoga and who is going with me ?
Response: yoga is with [alex] at [11am].
GLMP: your yoga is on [**thursday**]$_{\checkmark}$ with [**alex**]$_{\checkmark}$.
DF-Net: your yoga is on [**thursday**]$_{\checkmark}$ at [**5pm**]$_{\times}$.
COMET: your yoga is on [**thursday**]$_{\checkmark}$ at [**11am**]$_{\checkmark}$ with [**alex**]$_{\checkmark}$. (*More informative response.*)
Goal: [stanford_express_care, 6_miles, heavy_traffic, hospital, 214_el_camino_real]
Query: car i need a hospital , please send me a quick route to the one near me in the direction we are driving.
Response: stanford_express_care is on the way.
GLMP: stanford_express_care is [**2_miles**]$_{\times}$ away. (*Wrong distance info.*)
DF-Net: stanford_express_care is [**4_miles**]$_{\times}$ away. (*Wrong distance info.*)
COMET: stanford_express_care is [**6_miles**]$_{\checkmark}$ away.
Goal: [mountain_view, friday, clear_skies, 30f, 50f]
Query: what will be the highest_temperature on friday in mountain_view ?
Response: on friday , the highest_temperature in mountain_view is predicted to be [50f].
GLMP: it will be friday on mountain_view in highest_temperature. $\times$ (*Fail to retrieve the temperature info.*)
DF-Net: it will be highest_temperature on friday in mountain_view. $\times$ (*Fail to retrieve the temperature info.*)
COMET: on friday the temperature in mountain_view will be low of [**30f**]$_{\checkmark}$, and high of [**50f**]$_{\checkmark}$.
Goal: [teavana, 5_miles, moderate_traffic, coffee_or_tea_place, 145_amherst_st]
Query: get me directions to the nearest starbucks.
Response: there is **NO** starbucks locally.
GLMP: there is a starbucks [**1_miles**]$_{\times}$ away. (*Unfaithful response.*)
DF-Net: the nearest starbucks is teavana , it s [**1_miles**]$_{\times}$ away. (*Not fluent and wrong entities retrieved.*)
COMET: there is **NO** starbucks nearby , but [**teavana**]$_{\checkmark}$ is [**5_miles**]$_{\checkmark}$ away would you like directions there?

Table 6: Responses generated by our COMET, GLMP (Wu et al., 2019) and DF-Net (Qin et al., 2020) from the SMD dataset. Goal means the row that the user is queried. ✓ and × mean the right or wrong entity linked.

more fluent, informative, and accurate responses.

Specifically, in the first example, GLMP and DF-NET are lack of the necessary information "*11am*" or provide the wrong entity "*5pm*". But COMET can obtain all the correct entities, which is more informative. In the second example, our method can generated the response with the right "*distance*" information but GLMP and DF-Net can not. In the third example, GLMP and DF-Net can not even generate a fluent response, let alone the correct temperature information. But COMET can still perform well. The fourth example is more interesting: the user queries the information about "*starbucks*" which does not exist in the current KB. GLMP and DF-Net both fail to faithfully respond, whereas COMET can better reason KB to generate the right response and even provide an alternative option.

4 Related Work

Task-oriented dialogue systems can be mainly categorized into two parts: modularized (Williams and Young, 2007; Wen et al., 2017) and end-to-end (Eric and Manning, 2017). For the end-to-end task-oriented dialogue systems, (Eric and Manning, 2017) first explores the end-to-end method for the task-oriented dialogue systems. However, it can only link to the entities in the dialogue context and no KB is incorporated. To effectively incorporate the external KB, (Eric et al., 2017) proposes a key-value retrieval mechanism to sustain the grounded multi-domain discourse. (Madotto et al., 2018) augments the dialogue systems with end-to-end memory networks (Sukhbaatar et al., 2015). (Wen et al., 2018) models a dialogue state as a fixed-size distributed representation and uses this representation to query KB. (Lei et al., 2018) designs belief spans to track dialogue believes, allowing task-oriented dialogue systems to be modeled in a sequence-to-sequence way. (Gangi Reddy et al., 2019) proposes a multi-level memory to better leverage the external KB. (Wu et al., 2019) proposes a global-to-local memory pointer network to reduce the noise caused by KB. (Lin et al., 2019) proposes Heterogeneous Memory Networks to handle the heterogeneous information from different sources. (Qin et al., 2020) proposes a dynamic fusion mechanism to transfer the knowledge among different domains. (Yang et al., 2020) exploits the graph structural informa-

tion in KB and the dialogue. Other works also explore how to combine the Pre-trained Model (Devlin et al., 2018; Radford et al., 2019) with the end-to-end task-oriented dialogue systems. (Madotto et al., 2020a) directly embeds the KB into the parameters of GPT-2 (Radford et al., 2019) via fine-tuning. (Madotto et al., 2020b) proposes a dialogue model that is built with a fixed pre-trained conversational model and multiple trainable light-weight adapters.

We also notice that some existing works also combine Transformer with the memory component, e.g., (Ma et al., 2021). However, our method is distinguishable from them, since the existing works like (Ma et al., 2021) simply inject the memory component into Transformer. In contrast, inspired by the dynamic generation mechanism (Gou et al., 2020), the memory in COMET (i.e., the entity representation) is dynamically generated by fully contextualizing the KB and dialogue context via the Memory-masked Transformer.

5 Conclusion

In this work, we propose a novel COntext-aware Memory Enhanced Transformer (COMET) for the end-to-end task-oriented dialogue systems. By the designed Memory Mask scheme, COMET can fully contextualize the entity with all its KB and dialogue contexts, and generate the $(\mathcal{N}+1)$-tuple representations of the entities. The generated entity representations can further augment the framework and lead to better capabilities of response generation and entity linking. The extensive experiments demonstrate the effectiveness of our method.

Acknowledgements

We would like to thank the anonymous reviewers for their valuable comments.

References

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. MultiWOZ - a large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026, Brussels, Belgium. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

H. E., W. Zhang, and M. Song. 2019. Kb-transformer: Incorporating knowledge into end-to-end task-oriented dialog systems. In *15th International Conference on Semantics, Knowledge and Grids*, pages 44–48.

Mihail Eric, Lakshmi Krishnan, Francois Charette, and Christopher D. Manning. 2017. Key-value retrieval networks for task-oriented dialogue. In *Proceedings of the Annual SIGdial Meeting on Discourse and Dialogue*, pages 37–49, Saarbrücken, Germany. Association for Computational Linguistics.

Mihail Eric and Christopher Manning. 2017. A copy-augmented sequence-to-sequence architecture gives good performance on task-oriented dialogue. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics*, pages 468–473, Valencia, Spain. Association for Computational Linguistics.

Revanth Gangi Reddy, Danish Contractor, Dinesh Raghu, and Sachindra Joshi. 2019. Multi-level memory for task oriented dialogs. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3744–3754, Minneapolis, Minnesota. Association for Computational Linguistics.

Yanjie Gou, Yinjie Lei, Lingqiao Liu, Pingping Zhang, and Xi Peng. 2020. A dynamic parameter enhanced network for distant supervised relation extraction. *Knowledge-Based Systems*, 197:105912.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The efficient transformer. In *International Conference on Learning Representations*.

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 Annual Meeting of the Association for Computational Linguistics*, pages 1437–1447, Melbourne, Australia. Association for Computational Linguistics.

Zehao Lin, Xinjing Huang, Feng Ji, Haiqing Chen, and Yin Zhang. 2019. Task-oriented conversation generation using heterogeneous memory networks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing*, pages 4558–4567, Hong Kong, China. Association for Computational Linguistics.

Xutai Ma, Yongqiang Wang, Mohammad Javad Dousti, Philipp Koehn, and Juan Pino. 2021. Streaming simultaneous speech translation with augmented memory transformer. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 7523–7527. IEEE.

Andrea Madotto, Samuel Cahyawijaya, Genta Indra Winata, Yan Xu, Zihan Liu, Zhaojiang Lin, and Pascale Fung. 2020a. Learning knowledge bases with parameters for task-oriented dialogue systems. *arXiv preprint arXiv:2009.13656*.

Andrea Madotto, Zhaojiang Lin, Yejin Bang, and Pascale Fung. 2020b. The adapter-bot: All-in-one controllable conversational model. *arXiv preprint arXiv:2008.12579*.

Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. 2018. Mem2Seq: Effectively incorporating knowledge bases into end-to-end task-oriented dialog systems. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 1468–1478, Melbourne, Australia. Association for Computational Linguistics.

Feiping Nie, Heng Huang, Xiao Cai, and Chris H Ding. 2010. Efficient and robust feature selection via joint $l_{2,1}$-norms minimization. In *Advances in neural information processing systems*, pages 1813–1821.

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. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 133–142, Hong Kong, China. Association for Computational Linguistics.

Libo Qin, Xiao Xu, Wanxiang Che, Yue Zhang, and Ting Liu. 2020. Dynamic fusion network for multi-domain end-to-end task-oriented dialog. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 6344–6354, Online. Association for Computational Linguistics.

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.

Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. 2015. End-to-end memory networks. In *Proceedings of the International Conference on Neural Information Processing Systems*, page 2440–2448, Cambridge, MA, USA. MIT Press.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, undefinedukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the International Conference on Neural Information Processing Systems*, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. *International Conference on Learning Representations*.

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 International Conference on Computational Linguistics*, pages 3781–3792, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, Milica Gašić, Lina M. Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2017. A network-based end-to-end trainable task-oriented dialogue system. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics*, pages 438–449, Valencia, Spain. Association for Computational Linguistics.

Jason D Williams and Steve Young. 2007. Partially observable markov decision processes for spoken dialog systems. *Computer Speech & Language*, 21(2):393–422.

Chien-Sheng Wu, Richard Socher, and Caiming Xiong. 2019. Global-to-local memory pointer networks for task-oriented dialogue. In *International Conference on Learning Representations*.

Shiquan Yang, Rui Zhang, and Sarah Erfani. 2020. GraphDialog: Integrating graph knowledge into end-to-end task-oriented dialogue systems. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1878–1888, Online. Association for Computational Linguistics.

S. Young, M. Gašić, B. Thomson, and J. D. Williams. 2013. Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179.

A Appendices

A.1 Label Construction of Entity Linking

In practice, the datasets do not provide the golden linked entity. However, We could obtain a pseudo annotation by following (Qin et al., 2019) to use a distant supervision method. Specifically, we match the entities in the golden response against the entities in the memory $\mathcal{M}$ and use the matching result as the golden entity. For entities like "no_traffic", one may find matches in multiple rows. We resolve this ambiguity by choosing the entity from the row which has the most matches for all entities in the utterances.

A.2 Hyper-parameter Settings

Hyper-parameter	SMD	Multi-WOZ 2.1
Batch Size	32	16
Hidden Size	512	512
Embedding Size	512	512
#Layer of Dialogue Enc.	6	6
#Layer of Response Dec.	6	6
#Layer for Memory	3	3
#Head	8	8
Learning Rate	0.0001	0.0001
KB Mask Prob.	0.2	0.05
Dropout Prob.	0.1	0.1

Table 7: Hyper-parameters used in the two datasets.

We follow (Wu et al., 2019) to randomly mask a small number of entities into an unknown token to improve the generalization of our model. Besides, in the sketch generation and entity linking stages, we also use the label smoothing to regularize the model. The hyper-parameters such as dropout rate are tuned over the development set by grid search (*Entity F1* for both datasets). The model is implemented in PyTorch. The hyper-parameters used in two datasets are shown in Tab. 7.

A.3 Implementation Details of Other KB Representations with Transformer

To further compare the different methods of representing KB with our method, we also adopt the triplet, row-entity, and graph representation to replace our contextualized entity representation, where we keep the other parts of COMET unchanged.

Specifically, for the triplet representation, we follow (Madotto et al., 2018; Wu et al., 2019; Qin et al., 2020) to implement Transformer+Triplet, where the entity representation is the sum of the subject, relation, and object. Besides, the multi-hop reasoning (Sukhbaatar et al., 2015) is leveraged to further boost the performance. For the row-ent representation, we refer to (Gangi Reddy et al., 2019; Qin et al., 2019) to implement Transformer+Row&Ent, where Bag-of-word embedding and entity-type embedding are used for the row-level representation and entity-level representation. Besides, the row-level representation and entity-level representation are hierarchically queried, where the distribution of the entity-level embedding is used for the response generation. For the graph representation, we adopt the memory part of GraphDialog (Yang et al., 2020) to implement Transformer+Graph, where the entity embedding is further augmented by Graph Neural Networks (Veličković et al., 2018). Besides, the last hop of the triplet and graph representation, and the entity-level representation of Row&Entity representation will be also used to adaptively fuse the information of KB in the Decoder of COMET. More details can be found in the aforementioned papers.