Supplementary File for Knowledge-Consistent Dialogue Generation with Knowledge Graphs

Anonymous Author(s)
Affiliation
Address
email

A Discussion

Limitation As briefly discussed in Section G, our work is limited in multiple dimensions primarily in terms of dataset, retrieval, and generation. First, the benchmark dataset is limited. Despite the fact that there are several public Knowledge Graph (KG) available [22, 2], only one dataset [14] provides both the diverse set of dialogue and the corresponding large-scale KG. This circumstance may limit the rigorous evaluation of our framework's adaptability in various settings. Future work may study applying our approach for a wider range of dialogue datasets based on Wikipedia [4] by leveraging existing public large-scale KG such as Wikidata [22]. Second, the search space for retrieving context-relevant subgraphs can be expanded. Our SURGE framework now runs on a 1-hop KG that is rooted to entities in the given dialogue history. Finding the entity within the text, on the other hand, necessitates precise named entity extraction and entity linking. Therefore, future work may investigate extending our approach to a framework that can retrieve the context-relevant subgraph among entire KG instead of 1-hop KG. Third, there is still room for improvement in generation quality since we generate knowledge-enhanced responses with a small-scale Pre-trained Language Model (PLM) for efficiency. Such PLMs occasionally fail to generate natural sentences with a high quality [17]. Future work could aim to improve generation quality using a small-scale PLM.

Broader Impact Our proposed knowledge-grounded dialogue generation model is essential for designing user-friendly real-world AI systems. Among various types of dialogue generation models, knowledge-grounded dialogue models are trained to interact with users and convey factual information to users in natural languages. Their conversational features can be adapted to any user interfaces that connect the bilateral interaction between human and computer. We believe that the conversational interfaces can enhance the users' experiences and reduce the users' efforts in learning how to use the systems. However, knowledge-grounded dialogue models can become vulnerable to generating offensive, harmful, or misinformation responses depending on the users or data. When deploying the models in the real world, in addition to generating realistic responses, they also need to be robust to adversarial feedback from malicious users and biases inherited in pre-training or training corpus, or else they could malfunction. Along with the quantitative and qualitative evaluations on generated responses, it is worthwhile to examine robustness of the dialogue models.

B Notations

We organize the notations we used for formally describing our method in Table 1.

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

Table 1: A list of notations that we used for defining our method.

$\mathcal{V}$	pre-defined vocabulary of tokens for pre-trained language models (text)
$\mathcal{E}$	pre-defined vocabulary of entities (symbol)
$\mathcal{R}$	pre-defined vocabulary of relations (symbol)
$\mathtt{a}, \ldots \mathtt{z}$	knowledge graph symbols written in typewrite font
$\boldsymbol{x}$	input sequence (vector)
$x_1, \ldots, x_N$	input tokens (scalar)
$\boldsymbol{y} = [y_1, \ldots, y_T]$	output sequence and tokens
$\mathcal{G}$	multi-relational graph, such as knowledge graph
$\mathcal{Z}$	retrieved subgraph: $\mathcal{Z} \subset \mathcal{G}$
z	triplet (edge): $z \in \mathcal{Z}$
q_e	mapping function of entity symbol to sequence of tokens
q_r	mapping function of relation symbol to sequence of tokens
$q(\cdot)$	text representation function for retrieval
$d(\cdot)$	triplet representation function for retrieval
$\mathtt{Enc}$	Transformer Encoder
$\mathtt{Dec}$	Transformer Decoder
f	token (word) embedding function
θ	generator parameter
ϕ	retriever parameter
$\boldsymbol{\psi}$	set encoding function
$\boldsymbol{\beta}$	perturbation function
π	set permutation
n	the number of triplets in a retrieved subgraph $\mathcal{Z}$
k	the number of samples in a marginalization term
$\boldsymbol{z}$	encoder hidden state (single token)
$\boldsymbol{Z}$	encoder hidden states (sequence of tokens)
$\boldsymbol{h}$	decoder hidden state (single token)
$\boldsymbol{H}$	decoder hidden states (sequence of tokens)
$\boldsymbol{X}$	input embeddings after token embedding function (sequence)
$\boldsymbol{Y}$	output embeddings after token embedding function (sequence)

C Proofs

In this section, we first show that a naïve encoding function $\boldsymbol{\psi}$ in Section 3.4 is neither permutation invariant nor relation inversion invariant, formalized in Proposition C.1. After that, we prove that our invariant and efficient encoding function $\boldsymbol{\psi}^*$ with graph-conditioned token embedding perturbation is both permutation invariant and relation inversion invariant, formalized in Proposition C.2.

Proposition C.1. *A naïve encoding function $\boldsymbol{\psi}$ is neither permutation invariant nor relation inversion invariant.*

Proof. We prove this by contradiction.

Suppose $\boldsymbol{x} = [x_1, \ldots, x_n]$ and $\mathcal{Z} = \{(\mathtt{a}, \mathtt{d}, \mathtt{b}), (\mathtt{b}, \mathtt{e}, \mathtt{a}), (\mathtt{a}, \mathtt{d}, \mathtt{c})\}$. Moreover, let $\mathcal{Z}' = \{(\mathtt{b}, \mathtt{e}, \mathtt{a}), (\mathtt{a}, \mathtt{d}, \mathtt{b}), (\mathtt{a}, \mathtt{d}, \mathtt{c})\}$ be one of permutations of $\mathcal{Z}$ with the permutation order $\pi = (2, 1, 3)$.

From the definition of naïve encoding, $\boldsymbol{\psi}(\boldsymbol{x}, \mathcal{Z}) = [\boldsymbol{a}, \boldsymbol{d}, \boldsymbol{b}, \boldsymbol{b}, \boldsymbol{e}, \boldsymbol{a}, \boldsymbol{a}, \boldsymbol{d}, \boldsymbol{c}, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]$ and $\boldsymbol{\psi}(\boldsymbol{x}, \mathcal{Z}') = [\boldsymbol{b}, \boldsymbol{e}, \boldsymbol{a}, \boldsymbol{a}, \boldsymbol{d}, \boldsymbol{b}, \boldsymbol{a}, \boldsymbol{d}, \boldsymbol{c}, \boldsymbol{x}_1, ..., \boldsymbol{x}_n]$. Therefore, it is easy to notice that $\boldsymbol{\psi}(\boldsymbol{x}, \mathcal{Z}) \neq \boldsymbol{\psi}(\boldsymbol{x}, \mathcal{Z}')$, thus the naïve encoding is not permutation invariant.

We then show naïve encoding is not relation inversion invariant. Suppose $\mathcal{Z}'' = \{(\mathtt{a}, \mathtt{d}, \mathtt{b}), (\mathtt{b}, \mathtt{e}, \mathtt{a}), (\mathtt{c}, \neg\mathtt{d}, \mathtt{a})\}$, where $(\mathtt{a}, \mathtt{d}, \mathtt{c}) \in \mathcal{Z}$ is changed to its inverse relation $(\mathtt{c}, \neg\mathtt{d}, \mathtt{a})$. Then, $\boldsymbol{\psi}(\boldsymbol{x}, \mathcal{Z}'') = [\boldsymbol{a}, \boldsymbol{d}, \boldsymbol{b}, \boldsymbol{b}, \boldsymbol{e}, \boldsymbol{a}, \boldsymbol{c}, \neg\boldsymbol{d}, \boldsymbol{a}, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]$ that is different against $\boldsymbol{\psi}(\boldsymbol{x}, \mathcal{Z})$: $\boldsymbol{\psi}(\boldsymbol{x}, \mathcal{Z}) \neq \boldsymbol{\psi}(\boldsymbol{x}, \mathcal{Z}'')$. Therefore, the naïve encoding function is not relation inversion invariant.

In conclusion, from the above two counterexamples, we prove that a naïve encoding function $\boldsymbol{\psi}$ is neither permutation invariant nor relation inversion invariant. □

We now provide proof of the permutation invariance and the relation inversion invariance of our invariant and effective graph encoding $\boldsymbol{\psi}^*$, described in Section 3.4. Before starting the proof, we first revisit the permutation invariant property of graph neural networks that sum, mean and max operators are permutation invariant for the input set of AGGR. Thus, if we use sum, mean, or max for AGGR, then the token embedding perturbation function $\boldsymbol{\beta}$ naturally satisfies the permutation invariance property. In other words, $\boldsymbol{\beta}(\boldsymbol{X}, \mathcal{Z}) = \boldsymbol{\beta}(\boldsymbol{X}, \pi \cdot \mathcal{Z})$, where $\boldsymbol{X} = \tilde{\boldsymbol{\psi}}(\boldsymbol{x}, \texttt{SORT}(\texttt{ENT}(\mathcal{Z})))$ for any permutation π.

Proposition C.2. *Invariant and efficient encoding $\boldsymbol{\psi}^*$ is both permutation invariant and relation inversion invariant.*

Proof. Suppose $\boldsymbol{x} = [x_1, \ldots, x_n]$ and $\mathcal{Z} = \{(\texttt{a}, \texttt{d}, \texttt{b}), (\texttt{b}, \texttt{e}, \texttt{a}), (\texttt{a}, \texttt{d}, \texttt{c})\}$. We first consider the permutation invariance for any permuted set $\mathcal{Z}' = \pi \cdot \mathcal{Z}$. While $\mathcal{Z}$ and $\mathcal{Z}'$ can have different orders of elements thus the outputs of $\texttt{ENT}(\mathcal{Z})$ and $\texttt{ENT}(\mathcal{Z}')$ could be different, we always obtain the same output with the usage of the SORT operator for encoding. In other words, $\texttt{SORT}(\texttt{ENT}(\mathcal{Z})) = \texttt{SORT}(\texttt{ENT}(\mathcal{Z}'))$ holds due to the definition of the SORT operation in Eq. 5 of the main paper. Therefore, $\tilde{\boldsymbol{\psi}}(\boldsymbol{x}, \texttt{SORT}(\texttt{ENT}(\mathcal{Z}))) = \tilde{\boldsymbol{\psi}}(\boldsymbol{x}, \texttt{SORT}(\texttt{ENT}(\mathcal{Z}')))$ holds.

Further, since the token embedding perturbation function $\boldsymbol{\beta}(\cdot, \mathcal{Z})$ along with sum, max, or mean in AGGR is also permutation invariant with regards to any permutation on $\mathcal{Z}$, we conclude our invariant and efficient encoding $\boldsymbol{\psi}^*$ is permutation invariant.

We finally prove the relation inversion invariance property of $\boldsymbol{\psi}^*$. Suppose $\mathcal{Z}'' = (\mathcal{Z} \cup t') \setminus t$ where $t \in \mathcal{Z}$ is any triplet in a set and t' is inverse of t. Then, $\texttt{ENT}(\mathcal{Z}) = \texttt{ENT}(\mathcal{Z}'')$ that is trivial as $\texttt{ENT}(\mathcal{Z})$ returns the set of only unique nodes in $\mathcal{Z}$. Therefore, $\tilde{\boldsymbol{\psi}}(\boldsymbol{x}, \texttt{SORT}(\texttt{ENT}(\mathcal{Z}))) = \tilde{\boldsymbol{\psi}}(\boldsymbol{x}, \texttt{SORT}(\texttt{ENT}(\mathcal{Z}'')))$ correspondingly holds.

The remaining step to conclude the proof is to show the following equality: $\boldsymbol{\beta}(\cdot, \texttt{INV}(\mathcal{Z})) = \boldsymbol{\beta}(\cdot, \texttt{INV}(\mathcal{Z}''))$, to conclude that $\boldsymbol{\psi}^*(\boldsymbol{x}, \mathcal{Z}) = \boldsymbol{\psi}^*(\boldsymbol{x}, \mathcal{Z}'')$ from $\boldsymbol{\beta}(\tilde{\boldsymbol{\psi}}(\boldsymbol{x}, \texttt{SORT}(\texttt{ENT}(\mathcal{Z}))), \texttt{INV}(\mathcal{Z})) = \boldsymbol{\beta}(\tilde{\boldsymbol{\psi}}(\boldsymbol{x}, \texttt{SORT}(\texttt{ENT}(\mathcal{Z}''))), \texttt{INV}(\mathcal{Z}''))$. We note that $\texttt{INV}(\mathcal{Z}) = \texttt{INV}(\mathcal{Z}'')$, as INV makes any graph as bidirectional one by the definition in Eq. 6 of the main paper. Therefore, $\boldsymbol{\beta}(\cdot, \texttt{INV}(\mathcal{Z})) = \boldsymbol{\beta}(\cdot, \texttt{INV}(\mathcal{Z}''))$ holds, and the relation inversion invariance property of $\boldsymbol{\psi}^*$ holds.

□

D Experimental Setup

In this section, we introduce the detailed experimental setups for our models and baselines. Specifically, we describe the details on implementation, dataset, training and model in the following subsections of D.1, D.2, D.3 and D.4, one by one.

D.1 Implementation Details

We use the T5-small [17] as the base Pre-trained Language Model (PLM) for all experiments. For the pre-trained checkpoint, we use the version that the authors released. For all implementations, we use Pytorch [16]. To easily implement the language model, we use the huggingface transformers library [23].

Retriever Details In this paragraph, we describe the implementation details of our context-relevant subgraph retriever, including the triplet embedding and dialogue context embedding for the retriever.

For the dialogue history embedding function q, we use the existing pre-trained language model (PLM). Specifically, we use the encoder part of the T5-small model [17] and freeze the parameters of it not to be trained. We then instead add a Multi-Layer Perceptron (MLP) on top of it, to give a point-wise attention [1] to each token, whereby all tokens are not equally considered in the sentence encoding. Formally,

$$q(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i * \boldsymbol{z}_i, \qquad \boldsymbol{Z} = [\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n] = \texttt{Enc}(\boldsymbol{X}), \qquad \alpha_i = \frac{\exp(\texttt{MLP}(\boldsymbol{z}_i))}{\sum_{j=1}^{n} \exp(\texttt{MLP}(\boldsymbol{z}_j))} \; \forall i$$

where α_i is a scalar, and MLP is a Multi-Layer Perceptron consisting of two linear layers and ReLU nonlinearity.

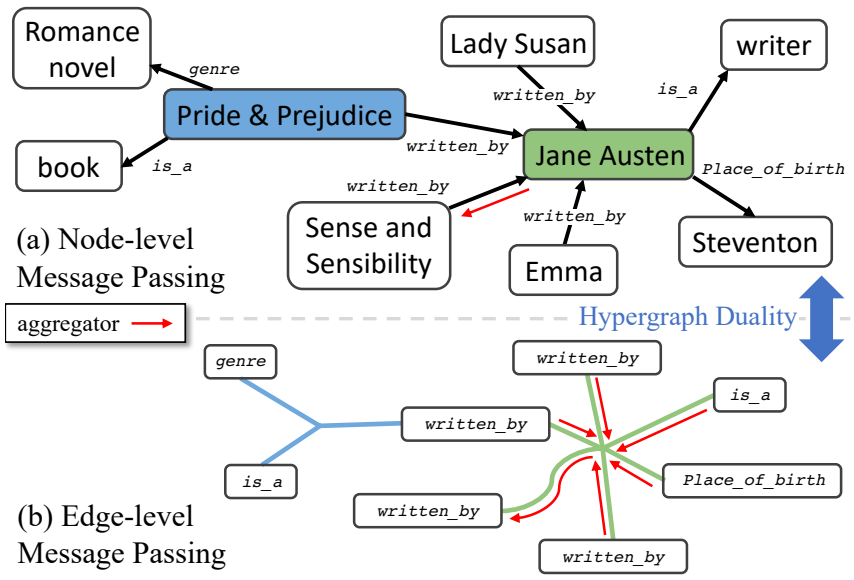


Figure 1: **Triplet Representation for Retrieval.** To represent each triplet with regards to its graph structure, we use the message passing on both nodes and edges. (a) Node-level Message Passing. To represent the entity *Sense and Sensibility*, the message from its neighbors – the entity *Jane Austen* – is aggregated. (b) Edge-level Message Passing. To represent the relation `written_by`, the messages from relations associated to a green hyperedge are aggregated. We do not draw self-loops and inverse edges for simplicity.

For obtaining triplet representations, we need to embed the entity (node) and relation (edge) into the latent space. Similar to the token embedding matrix used in PLMs, we can introduce the entity and relation embedding matrices. However, since the number of entities used in Freebase of OpendialKG [14] is too large compared to the number of tokens in T5 (100,814 vs 32,000) [17], it is inefficient to introduce the trainable entity embedding matrix for the retriever.

Thus, we instead reuse the contextualized representation from the PLM encoder, to embed each node if the corresponding entity exists in the dialogue context. Formally, suppose that there is a triplet $\{(\mathtt{e}_h, \mathtt{r}, \mathtt{e}_t)\}$ in the 1-hop subgraph $\mathcal{G}$, which satisfies the following condition: $q_e(\mathtt{e}_h) \subseteq \boldsymbol{x}$ or $q_e(\mathtt{e}_t) \subseteq \boldsymbol{x}$. If so, we can know the position of the mapped entity within the dialogue history: $[x_{start}, ..., x_{end}] = q_e(\mathtt{e}_h)$ from $q_e(\mathtt{e}_h) \subseteq \boldsymbol{x}$. Therefore, the node embedding for the entity $\mathtt{e}_h$ is obtained by $\mathtt{EntEmb}(\mathtt{e}_h) = \frac{1}{|q_e(\mathtt{e}_h)|} \sum_{i=start}^{end} \mathtt{Enc}(\boldsymbol{X})_i$ iff $q_e(\mathtt{e}_h) \subseteq \boldsymbol{x}$. For edge embedding, we use the trainable relation embedding matrix $\boldsymbol{R} \in \mathbb{R}^{|\mathcal{R}| \times 128}$ to represent the edge, since the number of relations is relatively small (1,357).

With our node and edge representations, we now focus on representing the triplet in Eq. 4 of the main paper for its retrieval. In particular, we use the Graph Neural Networks (GNNs) for encoding triplets, where we obtain the node representations from the Graph Convolutional Network (GCN) [9] that is a widely used architecture for representing the nodes with respect to their graph structures. However, for representing the edges, we use the Edge Hypergraph Graph Neural Network (EHGNN) used in Jo et al. [8], due to its simplicity but effectiveness for edge representations. We summarize our triplet representation in Figure 1.

Graph Encoder Details In this paragraph, we describe the implementation details of the token embedding perturbation function $\boldsymbol{\beta}$ used in our *Invariant and Efficient* graph encoding introduced in Section 3.4. To be aware of the relation of the graph over GNNs, we use the simplified version of CompGCN [20]. For architectural details, instead of using the different linear layers to distinguish the inverse relation from its opposite relation, we use the same linear layer. Also, we use subtraction as the specific composition operator for reflecting relations in CompGCN.

Then, we form the learnable affine transformation based on the aggregated representation from GNN layers, to perturb the token embeddings with respect to their graph information as in Eq. 7 of the

main paper. In particular,

$$\eta = \texttt{UPD}(f(a), \texttt{AGGR}(\{f(b), \texttt{r} \mid \forall \texttt{b} \in \mathcal{N}(\texttt{a}; \mathcal{Z})\})), \quad \boldsymbol{\gamma} = \texttt{MLP}_1(\eta), \quad \boldsymbol{\delta} = \texttt{MLP}_2(\eta),$$
$$\boldsymbol{\beta}(f(a), \mathcal{Z}) = (\mathbf{1} + \boldsymbol{\gamma}) * f(a) + \boldsymbol{\delta},$$

where $\texttt{MLP}_1$ and $\texttt{MLP}_2$ are learnable MLPs consisting of two linear layers with ReLU nonlinearity.

KQA Details In this paragraph, we describe the implementation details for our Knowledge-verifying Question Answering (KQA) introduced in Section 4. For building the QA dataset, we first gather the dialogue sessions where the gold response contains the entity from the whole OpendialKG dataset. Then, we extract the triplet from the given whole KG where the head entity is placed within the dialogue history and the tail entity is placed within the gold response. We build a QA training dataset based on the extracted triplets and a corresponding dialogue session. To diversify the training data, we replace the tail entity of each triplet with plausible candidate entities within KG and change the entity in the response following the changed entity on the triplet. As a result, we obtain the QA dataset size of 200k. We train the BERT-base [3] with the constructed QA dataset. We hold out 10% of data for validation and obtain the fine-tuned BERT model with 88.89 F1 score on the hold-out validation set. When we apply the fine-tuned QA model on the evaluation of the generated responses, we rebuild the QA evaluation set with the generated response instead of a gold response as illustrated in Figure 3 of the main paper.

D.2 Dataset Details

We mainly conduct experiments on **OpendialKG** [14], which provides the parallel dialogue corpus corresponding to the existing large-scale Knowledge Graph (KG) named Freebase [2]. The provided large-scale KG consists of total 1,190,658 fact triplets over 100,813 entities and 1,358 relations. This dataset is collected from 15K human-to-human role-playing dialogues, having multi-turns, from which we pre-process that each assistance response is the label and its corresponding dialogue history is the input. Although some of the data contain the gold knowledge that is useful for generating the response on the ongoing conversation, we found that 51% of data has no gold knowledge. To overcome this limitation, we additionally find entities from the dialogue history using the Named Entity Recognition module in spaCy[1], and then include the extracted entities' corresponding triplets in the KG to the dataset. Since the dataset does not provide the pre-defined data split, we randomly split sessions into train (70%), validation (15%), and test sets (15%).

D.3 Training Details

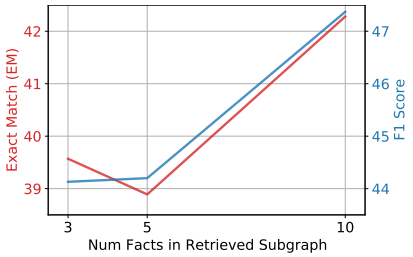
All experiments are constrained to be done with a single 48GB Quadro 8000 GPU. SURGE training needs 12 GPU hours. For all experiments, we select the best checkpoint on the validation set. We fine-tune the SURGE for 10 epochs on the training set, where we set the learning rate as 1e-4, weight decay as 0.01, learning rate decay warmup rate as 0.06, maximum sequence length for dialogue history as 256, maximum sequence length for knowledge as 128, and batch size as 24. For retrieval, we use the subgraph size n as 3, and sample size k for marginalization as 4. We use the AdamW [13] optimizer for training. For fairness, we apply the same training setting to all baselines if applicable.

D.4 Model Details

In this subsection, we describe the details of baselines and our models used in our experiments, as follows:

1. **No Knowledge**: This model is provided with only the dialog history. No knowledge is used to generate responses.
2. **Gold Knowledge**: This model is provided with the dialogue history along with its exact gold knowledge for the gold response. Thus, since this model uses such gold knowledge, we expect the results of it as the upper bound of the task.
3. **Space Efficient (series)**: This model is provided with all the knowledge which are related to the entities that appeared in the dialogue history [6], by matching the entities in the dialogue history and the entities in the KG. In particular, this model encodes the entities and their relations explicitly in the words in the encoder part.

[1] https://spacy.io/



Method	MRR	Hits@1	Hits@3	Hits@5	Hits@10	Hits@100
Random Retrieval	7.47	2.31	6.36	9.72	17.01	61.91
Sparse Retrieval (BM25)	7.17	2.22	6.23	8.98	16.36	56.88
Text-based Retrieval	24.39	15.46	26.97	33.94	43.25	72.36
SURGE (unsupervised)	19.66	9.55	22.46	29.81	41.09	69.35
SURGE (semi-supervised)	22.30	13.28	24.31	29.60	42.72	64.44

Figure 2: (Left:) Performances of our SURGE by varying the number of facts for retrieving the subgraph (i.e., varying the number of triplets in the subgraph) from three, to five, to ten, with EM and F1 scores of KQA as evaluation metrics. (Right:) We additionally report the knowledge retrieval performances, with MRR and Hits@K as evaluation metrics.

4. **Space Efficient (parallel)**: This model is mostly the same as the above model – space Efficient (series) – except the knowledge encoding part. Specifically, it encodes the entities in the words like the above, whereas, encoding the relation between entities in the segmentation block of the entities [6].
5. **EARL**: This model uses the RNN-based encoder-decoder architecture with the entity-agnostic representation learning [24], with all the provided knowledge associated with the entities in the dialogue history. Specifically, this model first calculates the probability of words obtained by encoding the entities in the KG, and then uses such probabilities to generate a word in the decoding phase.
6. **Random Retrieval**: This model is provided with entire facts from 1-hop subgraphs of entities that appeared in the dialogue history. However, instead of encoding all the knowledge in one-hop subgraph as in Space Efficient, this model randomly samples them, which are then used for generating responses.
7. **Sparse Retrieval** (BM25): This model is also provided with entire facts from 1-hop subgraphs of entities. To sample relevant facts to the dialogue history among the entire facts, this model uses BM25 [18] that is a sparse retrieval model. To be specific, let assume we have a dialogue history and its corresponding facts from 1-hop subgraphs of matched entities. Then, to run BM25, we first concatenate components of each fact consisting of two entities and one relation, and tokenize the dialogue history and the facts for obtaining corpus and queries, respectively, for BM25. After that, BM25 calculates the lexical overlapping score between the dialogue context (corpus) and the one-hop fact (query), from which we use the relevant facts having top-k scores by BM25.
8. **Text-based Retrieval**: This model uses a pre-trained language model as the triplet embedding function of the retriever similar to [7], instead of using GNN. Specifically, we consider each triplet as a single sentence (e.g, (Jane Austen, write, Susan) $\rightarrow$ "Jane Austen write Susan") and embed them with the pre-trained language model.
9. **SURGE (unsupervised)**: Our basic subgraph retrieval-augmented generation framework that is provided with entire facts from 1-hop subgraphs of entities. In particular, this model trains the structure-aware subgraph retriever without any guidance of the gold knowledge (i.e., ground truth knowledge for the dialogue history is not given). In other words, for the given dialogue context, this model implicitly learns to retrieve the context-relevant knowledge, and then generates the response with the retrieved knowledge.
10. **SURGE (semi-supervised)**: Our subgraph retrieval-augmented generation framework with semi-supervised learning of graph retrieval, with provided entire facts from 1-hop subgraphs of entities. Unlike the unsupervised version of SURGE, this model trains the retriever to select the gold knowledge if the dialogue context has such knowledge during training.
11. **SURGE (contrastive)**: Our full subgraph retrieval-augmented generation framework with the contrastive learning of graph-text modalities as well as the semi-supervised learning of graph retrieval, with provided entire facts from 1-hop subgraphs of entities. Unlike aforementioned frameworks of ours, this additionally enforces the model to faithfully reflect the retrieved knowledge in the input, to the generated response with contrastive learning.

Table 2: Experimental results on OpendialKG dataset with **BART-base** as the base PLM.

Method	KQA		BLEU				ROUGE			Unigram
	EM	F1	B-1	B-2	B-3	B-4	R-1	R-2	R-L	F1
No Knowledge *(BART-base)*	22.87	27.53	17.38	10.79	7.16	4.81	20.64	8.22	19.92	24.36
Space Efficient *(BART-base, Series)*	38.00	42.41	18.56	11.85	8.01	5.56	22.36	9.43	21.48	26.38
Space Efficient *(BART-base, Parallel)*	39.77	43.90	18.90	12.19	8.35	5.81	**22.63**	**9.79**	**21.76**	**26.79**
SURGE *(BART-base, semi-supervised, $n = 10$)*	41.85	45.75	**19.13**	**12.37**	**8.55**	**6.09**	21.81	9.26	20.97	26.41
SURGE *(T5-small, semi-supervised, $n = 3$)*	39.57	44.13	18.21	11.74	8.08	5.68	22.11	9.41	21.22	25.91
SURGE *(T5-small, semi-supervised, $n = 10$)*	**42.28**	**47.37**	18.04	11.70	8.11	5.75	22.08	9.49	21.13	26.02

E Additional Experiments

E.1 Varying the Number of Facts in Subgraphs

We experiment our SURGE framework with varying the number of facts in retrieval, which are then used in our graph encoding function to condition the encoded graph information for response generation. Specifically, in Figure 2, we report the EM and F1 scores measured by our KQA for our SURGE framework, with different numbers of facts within a retrieved subgraph: $n = [3, 5, 10]$. Note that, in this experiment, we only use the semi-supervised model without the contrastive loss. We expect that the performance of our SURGE will increase as we increase the number of facts within the retrieved subgraph, since the model can leverage more numbers of knowledge for response generation. As shown in Figure 2, we observe the significant performance improvements on using ten facts against using three and five facts, while the performance difference between the three and five is marginal. We suggest that this result should be interpreted with the retrieval results on the right side of Figure 2, where about 40% of retrieved subgraphs including the ten different facts contain at least one necessary knowledge, thus the generation performance is boosted according to the improvement in retrieval.

E.2 Discussions on Using Larger PLMs

Notably, we observe that the use of larger Pre-trained Language Models (PLMs) – three times more number of parameters compared to T5-small that we use – does not result in better performance for the knowledge-grounded dialogue task. Specifically, in Table 2, we report the experimental results of selected baselines and our SURGE semi-supervised model with BART-base [10] as the base PLM. We want to clarify that the BART-base model has 220M parameters, which is about **three times larger** than the number of parameters of the T5-small model (60M).

We first observe that BART-base shows decent performance without any knowledge (No Knowledge) compared to the no-knowledge case of T5-small, verifying that the larger PLM generally contains more factual knowledge within its pre-trained parameters. Moreover, BART-base obtains higher scores in the simple word overlap metrics such as BLEU [15] and ROUGE [12], whose results further confirm that a larger PLM can generate more natural or syntactically better sentences than the smaller one, thanks to its parameter size.

On the other hand, we find that BART-base is less suffered from the irrelevant knowledge issue (i.e., conditioning irrelevant knowledge for the given context when generating responses) than T5-small, therefore, the performance of *Space Efficient Encoding* on KQA is quite high. However, the use of BART-base does not result in significant improvement on the KQA metric for our SURGE framework. Moreover, ours with T5-small shows better performance than ours with BART-base in terms of KQA scores, when the number of facts within the retrieved subgraph is 10: $n = 10$. This result suggests that the quality of the generated response – having relevant knowledge to the given context – might depend on the performance of the subgraph retriever whose goal is to retrieve the context-relevant knowledge, rather than the inherent performance of PLMs.

E.3 Experimental Results on Another Dataset

In the main paper, we only experiment on OpendialKG dataset [14], since it is the largest and most realistic public datasets that provides both dialogues across diverse domains and corresponding large-scale Knowledge Graph (KG) [2]. To verify the effectiveness of our SURGE framework, the existence of the large-scale KG and the importance of relevant fact searching is important since we

Table 3: Experimental results on KOMODIS dataset with T5-small as the base PLM.

	BLEU				ROUGE			
	B-1	B-2	B-3	B-4	R-1	R-2	R-L	F1
No Knowledge	8.02	4.12	2.44	1.53	16.07	3.62	15.72	16.60
Random	9.45	5.30	3.48	2.47	17.60	4.50	17.20	18.57
Space Efficient *(Series)*	7.08	3.96	2.64	1.93	15.69	3.68	15.36	16.61
Space Efficient *(Parallel)*	7.71	4.45	3.00	2.20	16.61	4.16	16.27	17.65
SURGE (Ours)	**10.16**	**5.89**	**3.94**	**2.84**	**17.74**	**4.85**	**17.32**	**19.22**

Table 4: (Left:) Performance evaluation with the diversity metric named Distinct. (Right:) Ablation study results on GNN variants in our modules.

Method	Dist-1	Dist-2
No Knowledge	6.06	15.73
All Knowledge	9.67	24.45
SEE (Series)	8.49	21.77
SEE (Parallel)	8.78	22.70
EARL	5.15	16.46
Sparse Retrieval (BM25)	7.65	19.63
SURGE (semi-supervised)	**10.33**	**28.26**

	KQA		BLEU				ROUGE			Unigram
Method	EM	F1	B-1	B-2	B-3	B-4	R-1	R-2	R-L	F1
Eq 4. GCN → GAT	38.58	43.21	18.00	11.52	7.88	5.55	21.79	9.20	20.90	25.60
Eq 7. CompGCN → GCN	37.50	42.49	17.77	11.29	7.63	5.28	21.62	9.07	20.76	25.39
SURGE *(semi-supervised)*	**39.57**	**44.13**	**18.21**	**11.74**	**8.08**	**5.68**	22.11	9.41	21.22	25.91

focus on the real-world scenario where the response generation requires the relevant fact acquirement from the large-scale KG.

However, one can raise the question regarding the versatility of our method on other datasets. To alleviate the issue, we conduct additional experiments on another dataset named KOMODIS [5], which is also KG-based dialogue dataset. Compared to OpendialKG, KOMODIS does not provide the corresponding large-scale KG and most of responses do not require the knowledge. Therefore, we only measure the automatic evaluation to evaluate the performance of each method on KOMODIS dataset. In Table 3, we present the experimental results on the KOMODIS dataset. Results obviously show that our SURGE framework shows superior performance against baselines on the additional dataset. Therefore, we can conclude that our method can generalize to other datasets beyond the opendialKG dataset.

E.4 Diversity Evaluation

In the main paper, we evaluate model generation performance primarily on its quality. We measure the distinct metric [11], which is one of the most popular metrics for evaluating the diversity of the generative model, to evaluate the performance of each model in more diverse aspects. In Table 4 left, we report the performance of baselines and our models in distinct metric. Our SURGE framework generates more diverse responses than all other baselines, according to the results.

E.5 Ablations Studies on GNN Design Choices

We use two different types of Graph Neural Networks (GNN) in our SURGE framework. One is the Graph Convolutional Network (GCN) [9], which is used to embed each node entity on the entire 1-hop subgraph in the triplet embedding function d of the main paper Equation 4. Another is Composition-Based Multi-Relational Graph Convolutional Networks (CompGCN) [20], which is used to embed each entity by considering the relations between entities in the token embedding perturbation function $\boldsymbol{\beta}$ of the main paper Equation 7. In this subsection, we conduct ablation studies on both GNN design choices. First of all, we replace the GCN in Equation 4 with Graph Attention Network (GAT) [21] to validate the effect of the GNN design choices on the node embedding in the triplet embedding function. Then, we run experiments by changing CompGCN in Equation 7 to GCN to see how important the relationships are in the graph encoding. We present the results on Table 4 right. Results indicate that the use of GAT in Equation 4 does not have any impact on the performance a lot. However, the use of relation-aware GNN is highly important in effective and efficient graph encoding, since removing the relation awareness of GNN reduces the performance of our model a lot.

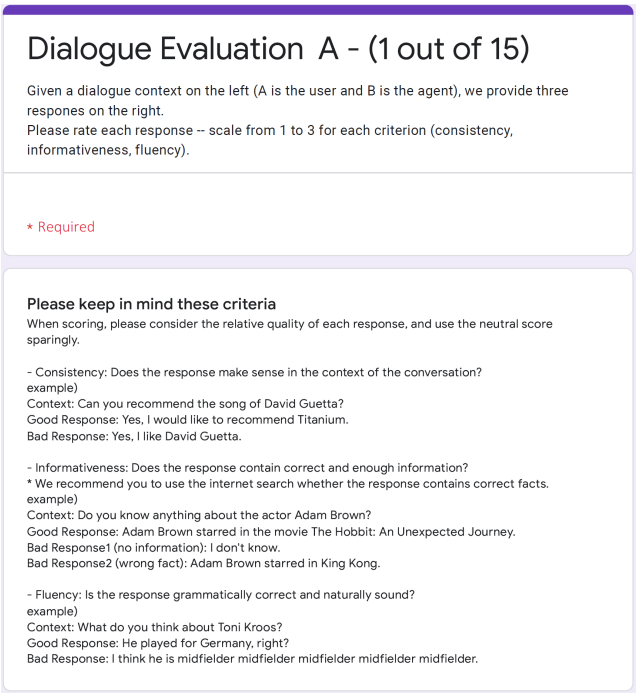


Figure 3: **Human Evaluation Instructions.** To measure the qualitative performances of the generated responses, annotators are provided with the following instruction on three criteria – consistency, informativeness, and fluency.

F Human Evaluation

In this section, we describe the details of human evaluation used in Section 5 of the main paper. We request the annotators to evaluate the responses generated from two baselines (i.e., ALL Knowledge and Space Efficient) and our SURGE framework in response to the given dialogue context, according to three criteria – consistency, informativeness, and fluency. Figure 3 is the instructions provided to each annotator. Specifically, regarding the consistency metric, we ask annotators to check whether the generated response makes sense in the context of the conversation. For informativeness, we ask annotators to check whether the response contains correct and enough information, whereby experiment participants are recommended to use the internet search, to check whether the response contains correct facts. In addition to this, we also provide the dialogue-related facts from Freebase as a reference for fact checking for annotators. For fluency, we ask annotators to check whether the response is grammatically correct and naturally sound.

G Retrieval and Generation Examples

In this section, we provide the examples for knowledge retrieval and response generation, for the given dialogue history.

Embeeding Space Visualization In Figure 4, we present a larger version of Figure 7 in the main paper. Specifically, we embed the hidden representations before the projection layer for each graph (star) and the embedding of the generated text (circle) through the dimensionality reduction using t-SNE [19]. As mentioned in the main paper, the visualization highlights that our SURGE framework with graph-text contrastive learning generates more distinct responses to different subgraphs, unlike the one without graph-text contrastive learning which shows less variety over responses even with different graphs.

Retrieval Examples We provide the retrieval examples of various models, such as random retrieval, sparse retrieval and our SURGE models. In particular, in the first (top) example of Figure 5, we are given a dialogue context in regard to books for Richard Maxwell, and baselines including random and BM25 retrievers select the facts associated to the entity Richard Maxwell, which are but irrelevant to

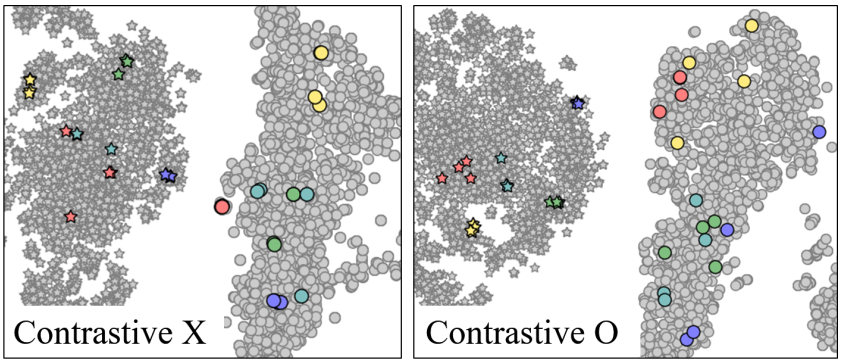


Figure 4: Large version of Figure 7 in the main paper. Stars indicate the embedding of graph and circles indicate the embedding of decoder hidden states (text), respectively.

the ongoing conversion, for example, (Richard maxwell, is-a Theatre director). Also, as shown in the second (bottom) example of Figure 5, we observe that the simple term-based matching model (i.e., BM25) cannot contextualize the current and previous dialogues, but retrieves the facts associated to frequent words, for example, song, which are less meaningful for the user's question. In contrast to baselines, as our SURGE framework trains a retriever in an end-to-end fashion, it first contextualizes the given dialogue context, and then accurately retrieves relevant knowledge.

Generation Examples We provide the generation examples from our model. To be specific, we provide the dialogue context along with its corresponding retrieved subgraph and generated response obtained from our SURGE framework. In Figure 6 and Figure 7, we provide the correct examples: our model retrieves a context-relevant subgraph, but also generates a factual response from retrieved knowledge. On the other hand, in Figure 8, we provide the failure cases. In particular, as shown in the first row of Figure 8, the fact in the knowledge graph could be ambiguous or inaccurate, as it defines the release year of the book – Wicked – as both 2008 and 2014. Moreover, we further provide the failure example on retrieval in the second row of Figure 8, where the user asks about the Bourne Legacy, while the dialogue agents retrieve the irrelevant knowledge to the question. Finally, we show the common problem in PLMs in the last row of Figure 8, where the generative model repeats the meaningless words at the end, while the retriever correctly selects the relevant knowledge.

Dialogue Context

A: Could you recommend any books written by Richard Maxwell?

Gold Knowledge

Richard maxwell, ~written_by, a tale of two cities

Random Knowledge

Richard maxwell, sibling, jan maxwell

Screenwriter, ~is-a, Richard maxwell

Theatre director, ~is-a, Richard maxwell

BM25 Knowledge

Richard maxwell, is-a, Theatre director

Screenwriter, ~is-a, Richard maxwell

Richard maxwell, organization founded, new york city players

Our Knowledge

Richard maxwell, ~written_by, a tale of two cities

Richard maxwell, sibling, Jan maxwell

Dialogue Context

A: I like Adam Levine.
B: OMG me too! I love that song Moves Like Jagger.
A: Yes, Love that too. It is really fun. Can you tell me more.
B: Did you know it's considered a power pop song?
A: No, I did'n. Do you know Love the way you Lie?

Gold Knowledge

Song, ~kind of composition, Love the way you lie

Love the way you lie, composer, Eminem

Random Knowledge

Blue monday, kind of composition, Song

The look of love, kind of composition, Song

Bad romance, kind of composition, Song

BM25 Knowledge

Song, ~kind of composition, This charming man

Behati prinsloo, ~spouse (or domestic partner), Adam levine

Song, ~kind of composition, safe & sound

Our Knowledge

g minor, ~key, Love the way you lie

Eminem, ~composer, Love the way you lie

Skylar grey, ~composer, Love the way you lie

Figure 5: Examples of the dialogue history with its corresponding gold knowledge as well as the retrieved knowledge from random retrieval and sparse retrieval baselines and from our SURGE framework. The retrieved fact is represented as the format of (head, relation, tail), where ∼symbol in the front of relation (i.e., ∼relation) in the retrieved knowledge denotes the inverse relation.

Dialogue Context

A: Do you like Rob Marshall? My friends have recommend his films.

Retrieved Knowledge

Nine, has_genre, Drama
Pirates of the Caribbean: On Stranger Tides, directed_by, Rob Marshall
Memoirs of a Geisha, directed_by, Rob Marshall

Generated Response

B: Yes, he wrote Pirates of the Caribbean: On Stranger Tides, Memoirs of a Geisha Nine and many others. Have you seen either of those?

Dialogue Context

A: Are there any works by Chris Cleave, which you can recommend me?
B: Certainly, wrote The Other Hand, Incendiary, and Old. Have you read them?
A: No, I haven't read The Other Hand out of the ones you mentioned. What genre is it?

Retrieved Knowledge

The Other Hand, release_year, 2008
2008, ~release_year, The Other Hand
The Other Hand, has_genre, Literary fiction

Generated Response

B: It's Literary fiction, and was released in 2008.

Dialogue Context

A: Who wrote Of Mice and Men?
B: It was written by John Steinbeck
A: Didn't he also write The Red Pony?
B: Yes He also wrote The Red Pony
A: Can you suggest other titles by him to read?

Retrieved Knowledge

Cannery Row, written_by, John Steinbeck
The Grapes of Wrath, written_by, John Steinbeck
Tortilla Flat, written_by, John Steinbeck

Generated Response

B: Yes he also wrote Tortilla Flat

Figure 6: Examples of the dialogue history with its corresponding retrieved knowledge and generated response from our SURGE framework. The fact is represented as the format of (head, relation, tail), where ∼symbol in the front of relation (i.e., ∼relation) in the retrieved knowledge denotes the inverse relation. In this example, we only provide the correct cases of both retrieval and generation.

Dialogue Context

A: Could you recommend some movies by director Simon Wood?

Retrieved Knowledge

The One That Got Away, written_by, Simon Wood
Simon Wood, ~written_by, The One That Got Away
Author, ~is-a, Simon Wood

Generated Response

B: Simon Wood directed The One That Got Away. Have you seen that?

Dialogue Context

A: I like David McCullough. Could you recommend any books of him?
B: Sure. He wrote The Greater Journey: Americans In Paris. Also, he wrote some documentary and Indie films.
A: Thank you for the information. When was The Greater Journey: Americans In Paris released?

Retrieved Knowledge

The Greater Journey: Americans In Paris, release_year, 2011
2011, ~release_year, The Greater Journey: Americans In Paris
Paris, written_by, Edward Rutherfurd

Generated Response

B: It was released in 2011. It's a great book.

Dialogue Context

A: I like Tony Tanner, what do you think?

Retrieved Knowledge

Sense and Sensibility, written_by, Tony Tanner
Tony Tanner, ~written_by, Mansfield Park
Mansfield Park, written_by, Tony Tanner

Generated Response

B: I think he is a great actor. He starred in Sense and Sensibility and Mansfield Park.

Figure 7: Examples of the dialogue history with its corresponding retrieved knowledge and generated response from our SURGE framework. The fact is represented as the format of (head, relation, tail), where ∼symbol in the front of relation (i.e., ∼relation) in the retrieved knowledge denotes the inverse relation. In this example, we only provide the correct cases of both retrieval and generation.

Dialogue Context

A: Do you know what books Jennifer L. Armentrout has written?
B: She wrote Wicked, White House Kiss and Wait For You. Are you interested in her other books as well?
A: No thanks. What genre is Wicked?
B: Wicked is a Play that won the Pulitzer Prize for Drama. Would you like more information about it?
A: Sure, sounds interesting

Retrieved Knowledge

Wicked, release_year, 2008
2008, ~release_year, Wicked
Wicked, release_year, 2014

Generated Response

B: It is a play written by Jennifer L. Armentrout and released in 2008.

Dialogue Context

A: I like the movie Armageddon.
B: Did you know Armageddon is written by Tony Gilroy?
A: Oh, didn't he also write The Bourne Legacy?

Retrieved Knowledge

Michael Clayton, starred_actors, Tony Gilroy
Michael Clayton, written_by, Tony Gilroy
Proof of Life, written_by, Tony Gilroy

Generated Response

B: Yes, he did. He also wrote Michael Clayton.

Dialogue Context

A: I like Deadpool or any movies like Deadpool.

Retrieved Knowledge

Deadpool, starred_actors, Ryan Reynolds
Deadpool, starred_actors, Stefan Kapicic
Deadpool, starred_actors, Ed Skrein

Generated Response

B: Deadpool is a great movie. Stefan Kapicic starred in it. Stefan Kapicic also starred in The Last Man and The Last Man.

Figure 8: Examples of the dialogue history with its corresponding retrieved knowledge and generated response from our SURGE framework. The fact is represented as the format of (head, relation, tail), where ∼symbol in the front of relation (i.e., ∼relation) in the retrieved knowledge denotes the inverse relation. In this example, we only provide the failure cases due to the problem on data (first row), retrieval (second row), and generation (third row).

References

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[2] Kurt D. Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, Vancouver, BC, Canada, June 10-12, 2008*, pages 1247–1250, 2008.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019. URL `https://doi.org/10.18653/v1/n19-1423`.

[4] Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. Wizard of wikipedia: Knowledge-powered conversational agents. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[5] Fabian Galetzka, Chukwuemeka Uchenna Eneh, and David Schlangen. A corpus of controlled opinionated and knowledgeable movie discussions for training neural conversation models. In Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asunción Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020*, pages 565–573. European Language Resources Association, 2020. URL `https://aclanthology.org/2020.lrec-1.71/`.

[6] Fabian Galetzka, Jewgeni Rose, David Schlangen, and Jens Lehmann. Space efficient context encoding for non-task-oriented dialogue generation with graph attention transformer. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 7028–7041. Association for Computational Linguistics, 2021.

[7] Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. Poly-encoders: Architectures and pre-training strategies for fast and accurate multi-sentence scoring. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL `https://openreview.net/forum?id=SkxgnnNFvH`.

[8] Jaehyeong Jo, Jinheon Baek, Seul Lee, Dongki Kim, Minki Kang, and Sung Ju Hwang. Edge representation learning with hypergraphs. *CoRR*, abs/2106.15845, 2021.

[9] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[10] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7871–7880. Association for Computational Linguistics, 2020.

[11] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models. In Kevin Knight, Ani Nenkova, and Owen

Rambow, editors, *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 110–119. The Association for Computational Linguistics, 2016. URL https://doi.org/10.18653/v1/n16-1014.

[12] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://aclanthology.org/W04-1013.

[13] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

[14] Seungwhan Moon, Pararth Shah, Anuj Kumar, and Rajen Subba. Opendialkg: Explainable conversational reasoning with attention-based walks over knowledge graphs. In Anna Korhonen, David R. Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 845–854. Association for Computational Linguistics, 2019.

[15] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*, pages 311–318. ACL, 2002.

[16] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035, 2019.

[17] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67, 2020.

[18] Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389, 2009.

[19] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. URL http://jmlr.org/papers/v9/vandermaaten08a.html.

[20] Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha P. Talukdar. Composition-based multi-relational graph convolutional networks. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[21] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[22] Denny Vrandecic and Markus Krötzsch. Wikidata: a free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85, 2014.

[23] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *EMNLP 2020 - Demos, Online, November 16-20, 2020*, pages 38–45, 2020. URL https://doi.org/10.18653/v1/2020.emnlp-demos.6.

[24] Hao Zhou, Minlie Huang, Yong Liu, Wei Chen, and Xiaoyan Zhu. EARL: informative knowledge-grounded conversation generation with entity-agnostic representation learning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 2383–2395. Association for Computational Linguistics, 2021.