Enhancing Dialogue Generation via Dynamic Graph Knowledge Aggregation

Chen Tang[1], Hongbo Zhang[2], Tyler Loakman[2], Chenghua Lin[2]* and Frank Guerin[1]
Department of Computer Science, The University of Sheffield, UK
Department of Computer Science, The University of Surrey, UK
{chen.tang,f.guerin}@surrey.ac.uk
{hzhang183,tcloakman1,c.lin}@sheffield.ac.uk

Abstract

Incorporating external graph knowledge into neural chatbot models has been proven effective for enhancing dialogue generation. However, in conventional graph neural networks (GNNs), message passing on a graph is independent from text, resulting in the graph representation hidden space differing from that of the text. This training regime of existing models therefore leads to a semantic gap between graph knowledge and text. In this study, we propose a novel framework for knowledge graph enhanced dialogue generation. We dynamically construct a multi-hop knowledge graph with pseudo nodes to involve the language model in feature aggregation within the graph at all steps. To avoid the semantic biases caused by learning on vanilla subgraphs, the proposed framework applies hierarchical graph attention to aggregate graph features on pseudo nodes and then attains a global feature. Therefore, the framework can better utilise the heterogeneous features from both the post and external graph knowledge. Extensive experiments demonstrate that our framework outperforms state-of-the-art (SOTA) baselines on dialogue generation. Further analysis also shows that our representation learning framework can fill the semantic gap by coagulating representations of both text and graph knowledge. Moreover, the language model also learns how to better select knowledge triples for a more informative response via exploiting subgraph patterns within our feature aggregation process. Our code and resources are available at https://github.com/tangg555/SaBART.

1 Introduction

Recent years have seen a surge of interest in developing chatbots with the facilitation of large-scale knowledge (Ni et al., 2022). As a highly expressive data format, Knowledge Graphs (e.g. ConceptNet and DBpedia), which include world facts, are considered to be a key factor in building an effective dialogue generation system (Zhou et al., 2018). In order to incorporate graph-structured knowledge, a range of Graph Neural Networks such as Graph Attention Networks (GATs) (Velickovic et al., 2017; Brody et al., 2021) and Graph Convolutional Networks (GCNs) (Kipf and Welling, 2016) have been proposed to learn representations of the topological structure of the knowledge graph via message passing between entities. In open-domain dialogue generation, these GNNs are further embedded into generative frameworks to feed graph knowledge features into the language models (LMs).

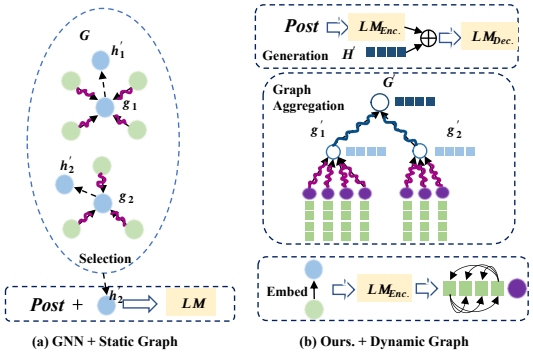

Figure 1: Conventional GNNs vs. Ours.

Despite prior success in leveraging graph knowledge with graph neural networks (GNN) (Zhou et al., 2018; Zhang et al., 2020), current generative frameworks are still hindered by the representation gap in the hidden space between the LMs and GNNs, which poses significant challenges in exploiting graph knowledge in the subsequent text decoding process. As illustrated in Figure 1, prior works using GNNs (Zhu et al., 2017; Ghazvininejad et al., 2018; Zhou et al., 2018; Zhang et al., 2020) tend to fuse the graph features by transforming them into text form and then feeding them into the language model, which acts as a "copy" mechanism. In other words, these networks run as a pipeline where the graph knowledge is firstly transformed into additional text to avoid the problem

*Corresponding author.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 4604–4616

of language model encoding brought about by the heterogeneous graph features. However, these separate encoding stages result in neural networks learning suboptimal representations of graph knowledge, which leads to information loss. With large-scale pretrained models such as GPT-2 (Radford et al., 2019), BART (Lewis et al., 2020) and T5 (Raffel et al., 2020) being widely adopted in recent advances in dialogue generation, the drawbacks that arise from incompatibility between GNNs and LMs becomes a more severe problem, prohibiting chatbot systems from leveraging graph structured data effectively.

In order to address the aforementioned challenges, we propose a novel representation learning framework to facilitate language understanding and generation, which permits effective incorporation of heterogeneous features via a dynamic graph knowledge aggregation mechanism. In contrast to existing works (Zhu et al., 2017; Ghazvininejad et al., 2018; Zhou et al., 2018; Zhang et al., 2020) which incorporate graph knowledge with conventional GNNs (causing inadequacies in representation learning), we propose to involve language models in both text and graph knowledge incorporation at all steps via hierarchically aggregating knowledge on a dynamic pseudo graph. During the knowledge aggregation process, knowledge triples are reorganised as shown in Figure 1 (b), where pseudo nodes are created to learn conceptual representations from original knowledge triples. Conceptual semantics are forced to coagulate into pseudo nodes, and finally merge into a condensed feature vector to fill the semantic gap of the encoded text features. Our approach for incorporating text and graph knowledge features can be adapted to all language models with an encoder-decoder architecture. In this study, we choose BART (Lewis et al., 2020), a SOTA language model for generation, as our language model in our experiments. This framework will hereinafter be referred to as SaBART (Subgraph-Aggregation BART).

During subgraph knowledge aggregation, the language model is involved in learning three levels of features: (1) Subword-level, where conceptual embeddings are connected to entity mentions within text; (2) Knowledge-level, where original triples are transformed by language encoding; and (3) Semantic-level, where the context vector encoded from text is involved in knowledge aggregation. This implies that the neural networks are able to access both the text and graph features during representation learning. The text and graph unified encoding process also avoids the information loss caused by the representation shift in vanilla GNNs, thus improving efficiency and efficacy. Extensive experiments demonstrate that our proposed framework significantly outperforms current SOTA baselines in dialogue generation. We also conduct in-depth analysis into the underlying mechanism of why our proposed approach better incorporates heterogeneous features. Our contributions can be summarised as follows:

- We propose a novel representation learning framework where graph and text features can be effectively aggregated via hierarchical knowledge aggregation on a dynamically constructed pseudo graph;
- We conduct a comprehensive set of experiments to demonstrate the effectiveness of our proposed approach, where our framework achieves SOTA performance on the commonsense knowledge graph enhanced dialogue generation dataset;
- We conduct in-depth experiments to analyse the improvement of representation learning on both graph and text knowledge, and investigate the mechanism to address this representation gap problem of learning heterogeneous features.

2 Related Works

In this section, we introduce related works by summarising recent advances in the knowledge enhanced dialogue generation task, as well as the SOTA approaches for injecting graph knowledge into generative frameworks.

Knowledge Enhanced Dialogue Generation As a data-driven approach, deep learning based chatbots rely on access to large amounts of knowledge (Zhao et al., 2020) to generate interesting and informative dialogues like humans. In order to realise a commonsense-aware and semantic-aware chatbot, more and more studies (Yu et al., 2022; Huang et al., 2022; Tang et al., 2022b) aim to consider external knowledge beyond pure dialogue exchanges to facilitate generation, where knowledge graphs containing topological structural knowledge are an important research direction to facilitate logical reasoning. In this study, we aim to improve

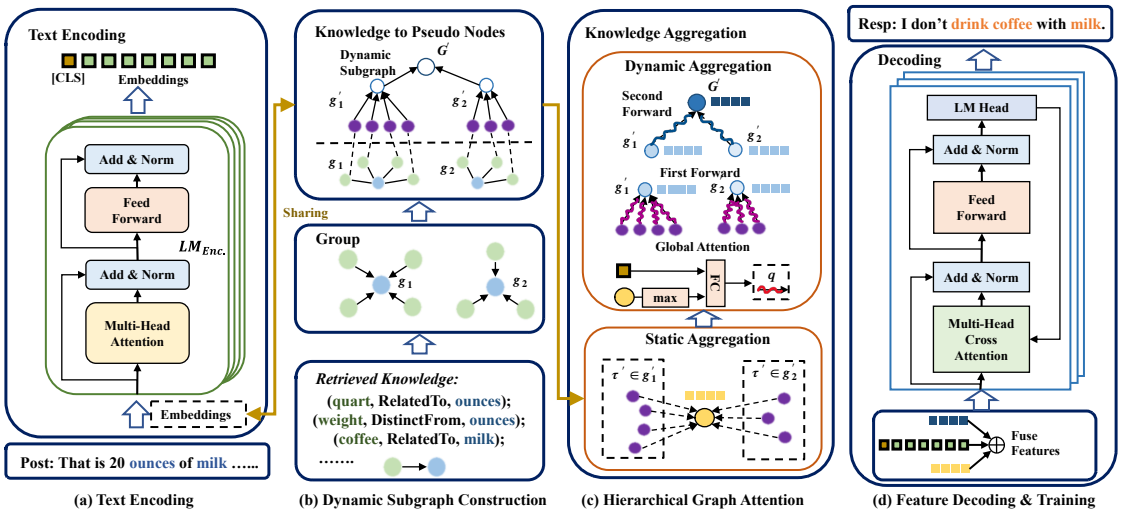

Figure 2: The overall framework of SaBART. We split the whole framework into four parts, which contain co-dependencies. For instance, the concepts and relations will create new sub-word embedding placeholders in language models after flattening into text sequences to make the graph and language models share low-dimensional representations.

the performance of neural frameworks through using additional knowledge graphs as external resources. Therefore, studies with external knowledge other than knowledge graphs, such as content planning (Tang et al., 2022c), retrieved documents (Yu et al., 2022), and mixed resources (Wu et al., 2022), are not directly compared in this paper. We believe our learning pattern for handling heterogeneous features can provide inspiration to other types of knowledge grounded conversational systems as well.

Injecting Graph Knowledge into Generative Frameworks With pre-training techniques being widely adopted, large-scale language models such as UniLM (Dong et al., 2019), GPT-2 (Radford et al., 2019), and BART (Lewis et al., 2020) have become the base models for various dialogue generation systems. However, these language models generally input only sequence formatted text, and cannot directly incorporate features of graph knowledge. Usually graph knowledge has to firstly be flattened into a sequence of tokens (Tang et al., 2022a), or encoded as feature vectors (Zhao et al., 2020) before being fed into a language model. Zhou et al. (2018) uses GRUs and two graph attention modules to select appropriate triples to incorporate into responses. In order to exploit the benefits of multi-hop knowledge, Zhang et al. (2020) adds an attention mechanism in a similar way to filter the appropriate knowledge. Finally, Tuan et al. (2019) propose a model which selects the output from a sequence-to-sequence model and a multi-hop reasoning model at each time step.

3 Methods

As illustrated in Figure 2, our approach aims to improve dialogue generation of language models by better incorporating heterogeneous features with an effective knowledge aggregation framework on retrieved knowledge triples.

3.1 Task Definition

In our task, the given input includes a textual post $X = \{x_1, x_2, ..., x_n\}$ where x_n denotes the n-th token, and a graph knowledge base $G = \{\tau_1, \tau_2, ..., \tau_k\}$. τ denotes a triple $\{h, r, t\}$, where h, r, and t refer to the head entity, the relation, and the tail entity, respectively. These triples represent the entities contained in the posts and reference responses, and the relations between them. Provided with these two kinds of input, the generation model is required to generate a response $Y = \{y_1, y_2, ..., y_m\}$ by modeling the conditional probability distribution $P(Y|X, G)$.

3.2 Dynamic Subgraph Construction

The graph knowledge is obtained by retrieving concept triples from ConceptNet, which are contained in the posts and reference responses. Our knowledge retrieval process is implemented by word matching (concepts in ConceptNet take the form of single words) and rule filtering to collect knowledge triples, which resembles the strategy of Zhou

et al. (2018). This process involves recognising relevant conceptual entities contained in the input post, and retrieving directly connected concept entities in the responses, with the goal of exploiting these entities in output responses.[1] Therefore, during knowledge retrieval, the retrieved knowledge triples are grouped according to the mentions of conceptual entities from the posts. For example, the post given in Figure 2 has "milk" recognised as an entity mention, which in turn retrieves relevant triples, e.g. (`coffee`, `RelatedTo`, `milk`). First of all, we group the retrieved knowledge triples as follows:

$$\text{ent}_1, \text{ent}_2, ..., \text{ent}_n \in X \cup G \quad (1)$$

$$g_i = \{\tau_1, \tau_2, ..., \tau_j\} \quad \text{s.t. ent}_j \in \tau \quad (2)$$

where triples τ containing the same conceptual entity ent_i are grouped as a subgraph g_i for the post. In contrast to existing works (Zhu et al., 2017; Ghazvininejad et al., 2018; Zhou et al., 2018; Zhang et al., 2020) that encode knowledge and select triples on separate subgraphs (leading to biased and incomplete feature learning), we propose to reconstruct G with pseudo nodes, so that pseudo nodes can dynamically connect each g_i to form a global graph:

$$pseu_{\tau_j} = \text{LM}_{emb}(F_{flatten}(\tau_j)) \quad (3)$$

$$F_{flatten}(\tau_j) = [x_j^h, x_j^{r_0}, x_j^{r_1}, x_j^t] \quad (4)$$

$$F_{insert}(F_{flatten}(\tau_j)|\tau_j \in G) \xrightarrow{emb} \text{LM}_{emb} \quad (5)$$

where $F_{flatten}$ flattens the triple of (h, r, t) to a text sequence, e.g., (coffee, RelatedTo, milk) will be flattened to "coffee related to milk". In ConceptNet, h and t consist of single words, and r is the relation of the two words. These words are transformed into BPE (byte-pair encoding) pieces (Lewis et al., 2020), and distinctively inserted into the sub-word embedding layer of the LM. This is performed in order to learn the semantics of both the entity mentions in the post, as well as the topological structure of the graph $pseu_{\tau_j} \in \mathbb{R}^{WE}$ (E denotes the size of the word embeddings), which constitutes the representation of the whole triple. We replace the original triples of G with the basic pseudo nodes (in purple) , where W is the word length of flattened τ_j, and C is the embedding size of the LM. On top of the basic pseudo nodes, hierarchical pseudo layers are created to connect all subgraphs:

$$g_i^{'} = \{\tau_1^{'}, \tau_2^{'}, ..., \tau_j^{'}\} \quad \text{s.t. } \tau_j \in g_i \quad (6)$$

$$\tau_j^{'} = \{pseu_{\tau_j}, r_a, pseu_{g_i}\} \quad (7)$$

where $g_i^{'}$ denotes the subgraph rebuilt by pseudo nodes $pseu_{\tau_{1<j<=|g_i|}} \in \mathbb{R}^{WC}$. They are connected to $pseu_{g_i}$ with a relation r_a, whose weight is calculated with an attention mechanism introduced in Sec. 3.3.

$$G^{'} = \{T_1, T_2, ..., T_k\} \quad (8)$$

$$T_k = \{pseu_{g_k}, r_a, pseu_G\} \quad (9)$$

Here $pseu_G \in \mathbb{R}^{WC}$ is the root node representing the features of the whole graph $G^{'}$. $pseu_G$ as the new pseudo knowledge graph is the set of the aforementioned pseudo triples transformed from the original triples.

3.3 Hierarchical Knowledge Aggregation

Instead of learning graph features by message passing on graph nodes, we implement a novel representation learning framework, where we train the neural network to learn the global features from the whole graph by hierarchically aggregating features through pseudo nodes as shown in Figure 2(c). Firstly, we encode features of the post to obtain a semantic vector $\mathbf{H}^{CLS} \in \mathbb{R}^{1\times E}$, and the embeddings of input tokens $\mathbf{H}^X$:

$$\begin{aligned} LM_{enc}(X) &= [\mathbf{H}^{CLS}; \mathbf{H}^X] \\ &= [emb_0^c; emb_1, emb_2, ...] \end{aligned} \quad (10)$$

where the context information of the post $\mathbf{H}^{CLS}$ will be used as context from the post, and involved in aggregating features on graph knowledge (as the query vector in the attention mechanism). Subsequently, a series of feature incorporation procedures will be conducted on our constructed graph of pseudo nodes.

3.3.1 Aggregation on Static Graphs

In §3.2, the original retrieved triples have been transformed into the set of pseudo nodes $pseu_{\tau_j}$. To obtain the representation of the graph, we directly aggregate the node features by calculating their mean value, which is inspired by the work of Tang et al. (2021) in calculating global representations.

$$\epsilon = \frac{\sum_{\tau_j \in G} pseu_{\tau_j}}{|G|} \quad (11)$$

[1] The concepts and their relations come from ConceptNet `https://conceptnet.io/`.

where $\epsilon \in \mathbb{R}^{WE}$ denotes the semantic representation of all triples. Because every node has the same weight when contributing to the global representation, ϵ will be carried into the following dynamic graph aggregation to obtain a better graph representation feature for response generation.

3.3.2 Aggregation on Dynamic Graphs

The aggregation process on the dynamic knowledge graph has a sequential forward procedure as follows.

First Forward Layer. In the first step, we calculate the features of $pseu_{g_i} \in g_i$:

$$\text{Update}(pseu_{g_i}) = \sum_{j=1}^{|g_i^{'}|} a_{ji}^{g_i^{'}} pseu_{\tau_j} \tag{12}$$

$$a_{ji}^{g_i^{'}} = \frac{\exp(\beta_{ji}^{g_i^{'}})}{\sum_{j=1}^{|g_i^{'}|} \exp(\beta_{ji}^{g_i^{'}})} \tag{13}$$

$$\beta_j^{g_i^{'}} = \mathbf{W}^{g_i^{'}}[pseu_{\tau_j}; q]^{\mathrm{T}} \tag{14}$$

$$q = FC([\mathbf{H}^{CLS}; \epsilon^{'}]) \tag{15}$$

$$\epsilon^{'} = max_{pool}(\epsilon) \tag{16}$$

where $\tau_j^{'} \in g_i^{'}$ all include $pseu_{g_i}$ as the tail node (cf. Equation 6); i denotes the i-th entity mention in the post; and j denotes the j-th triple related to the mention. $\mathbf{W}^{g_i^{'}} \in \mathbb{R}^{1\times(W+1)E}$ is a trainable parameter multiplying the concatenation of the node representation and the context feature vector. $a_{ji}^{g_i^{'}}$ is an attention score to aggregate features of $pseu_{\tau_j}$ into the updated $pseu_{g_i}$. $q \in \mathbb{R}^{2E}$ is the query vector of the attention mechanism. FC is a fully connected neural network, and max_{pool} is the max-pool function transforming $\epsilon \in \mathbb{R}^{WE}$ to $\epsilon^{'} \in \mathbb{R}^{E}$.

Second Forward Layer. Similarly, when our model attends to $G^{'}$, we update the features with $pseu_{g_k}$ obtained in the first step:

$$\text{Update}(pseu_G) = \sum_{k=1}^{|G^{'}|} a_k^{G^{'}} pseu_{g_k} \tag{17}$$

$$a_k^{G^{'}} = \frac{\exp(\beta_k^{G^{'}})}{\sum_{j=1}^{|G^{'}|} \exp(\beta_k^{G^{'}})} \tag{18}$$

$$\beta_k^{G^{'}} = \mathbf{W}^{G^{'}}[pseu_{g_k}; q]^{\mathrm{T}} \tag{19}$$

where $\mathbf{W}^{G^{'}} \in \mathbb{R}^{1\times(W+1)E}$ is a trainable parameter, and the final $pseu_G$ represents the global features aggregated by G. The feature vector q is the same as the one in the first forward layer, which acts as the global context of both the post and static knowledge graph.

Datasets	Train	Val	Test
Conversational # Pairs	3,384,185	20,000	10,000
Vocabulary Size	39,674	27,115	18,036
Retrieved # Entities	108,410	27,135	20,125
Retrieved # Triples	120,848	110,952	86,940
Avg. # Entities in Input Posts	92.25	94.64	92.84
Avg. # Entities in Output Responses	2.33	2.31	2.33
Avg. # Subgraphs in # Pairs	5.77	6.47	5.81
Avg. # Triples in # Pairs	105.43	108.12	106.04

Table 1: Data statistics of the commonsense dialogue generation dataset. Retrieved entities and triples show the unique entities and triples contained in the dataset. The definition of a subgraph refers to subsection 3.2

3.4 Inference and Training

To auto-regressively generate responses, the language model predicts each token y_t at time step t:

$$Y_t = [y_1, y_2, ..., y_t] \quad \text{s.t.} \quad t > 0 \tag{20}$$

$$p_{y_t} = \text{softmax}(\mathbf{H}^{dec}\mathbf{W}^{res}) \tag{21}$$

$$\mathbf{H}^{dec} = \text{LM}_{dec}(\mathbf{H}^{enc}, Y_{t-1}) \tag{22}$$

$$\mathbf{H}^{enc} = [pseu_G; \epsilon; \mathbf{H}^{CLS}; \mathbf{H}^{X}] \tag{23}$$

where $\mathbf{H}^{enc} \in \mathbb{R}^{L\times E}$ and $\mathbf{H}^{dec} \in \mathbb{R}^{1\times E}$ are outputs of encoders and decoders; and L denotes the size of the concatenated feature vector. The dimension of $pseu_G$ here is transformed to $\mathbb{R}^{W\times E}$, so that $pseu_G$, $\mathbf{H}^{CLS}$ and $\mathbf{H}^{X}$ can be concatenated at the first dimension. $W^{res} \in \mathbb{R}^{L\times E}$ is a trainable parameter denoting the LM head in Figure 2(d); and E denotes the size of the word embeddings. Finally, we train the whole framework with the loss function as follows:

$$\mathcal{L} = -\frac{1}{N}\sum_{n=1}^{N} \log P(Y|X, G) \tag{24}$$

where N denotes the size of the test data, and $\mathcal{L}$ is the cross entropy of predicted response tokens and those of the golden responses.

4 Experiment

4.1 Experimental Setup

Dataset. We process the dataset provided by Zhou et al. (2018) for the following experiments, where the train/val/test datasets are split into sizes

Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	NIST-1	NIST-2	NIST-3	NIST-4	METEOR
Seq2Seq	0.1702	0.0579	0.0226	0.0098	1.0230	1.0963	1.1056	1.1069	0.0611
MemNet	0.1741	0.0604	0.0246	0.0112	1.0975	1.1847	1.1960	1.1977	0.0632
CopyNet	0.1589	0.0549	0.0226	0.0106	0.9899	1.0664	1.0770	1.0788	0.0610
CCM	0.1413	0.0484	0.0192	0.0084	0.8362	0.9000	0.9082	0.9095	0.0630
UniLM	0.2019	0.0730	0.0305	0.0138	1.3562	1.4919	1.5082	1.5101	0.0796
ConceptFlow	0.2451	0.1047	0.0493	0.0246	1.6137	1.7956	1.8265	1.8329	0.0942
SaBART (ours)	**0.3298**	**0.2113**	**0.1467**	**0.0945**	**2.9226**	**3.8386**	**4.0763**	**4.1121**	0.1674
- w/o dy-agg	0.2967	0.1909	0.1322	0.0846	2.3889	3.1635	3.3618	3.3914	0.1647
- w/o st-agg	0.2927	0.1880	0.1299	0.0831	2.3712	3.1593	3.3623	3.3928	**0.1684**
- w/o kg	0.1446	0.0578	0.0285	0.0155	1.0381	1.1653	1.2245	1.2327	0.0931

Table 2: Automatic evaluation on referenced metrics used in the task of open domain dialogue. The best performing model is highlighted in **bold**. $-w/o$ stands for the ablated model. ***dy-agg*** denotes the aggregation on the dynamic graph, where Equation 23 will exclude the input of $pseu_G$. ***st-agg*** denotes the aggregation on static graph, where every element related to ϵ will be excluded, including the q vector used in ***dy-agg***. ***-w/o kg*** denotes the model learning the input without external graph knowledge (equivalent to vanilla BART).

of 3,384,185/20,000/10,000, respectively.[2] The statistics of the data are shown in Table 1. From the table, it can be observed that the average statistics of entities, subgraphs and triples in these three splits are very close, implying that the data samples are fully shuffled to make the experiment fair.

Baselines. We select several competitive baselines for comparison, including: **Seq2seq** (Sutskever et al., 2014), **MemNet** (Ghazvininejad et al., 2018), **CopyNet** (Zhu et al., 2017), **UniLM** (Dong et al., 2019), **BART** (Lewis et al., 2020), **CCM** (Zhou et al., 2018), and **ConceptFlow** (Zhang et al., 2020). In particular, UniLM and BART are SOTA pre-trained models for generation tasks, whilst ConceptFlow is the SOTA model for our task.[3]

Evaluation Metrics. We adopt the metrics of BLEU (Papineni et al., 2002), NIST (Doddington, 2002), METEOR (Lavie and Agarwal, 2007), Dist, and Ent (Zhang et al., 2018) for evaluation. BLEU, NIST, and METEOR are calculated between generated responses and golden responses, whilst Dist and Ent (calculating word distinction by n-grams) are calculated within generated responses. We also conduct further experiments to evaluate the efficiency and efficacy of incorporating external knowledge by counting the entities from the post used in the generated responses.

Model	Dist-1	Dist-2	Ent-4
Seq2Seq	0.0123	0.0525	7.665
MemNet	0.0211	0.0931	8.418
CopyNet	0.0223	0.0988	8.422
CCM	0.0146	0.0643	7.847
UniLM	0.0189	0.0755	9.599
Conceptflow	0.0223	0.1228	10.270
SaBART (ours)	0.0598	0.2798	9.456
- w/o dy-agg	0.0607	0.2739	5.388
- w/o st-agg	**0.0616**	**0.2816**	9.916
- w/o kg	0.0055	0.1752	**10.400**

Table 3: Automatic evaluation on unreferenced metrics.

4.2 Implementation Details

Our framework is mainly implemented with Pytorch[4] and Pytorch-lightning, and we select BART (Lewis et al., 2020) as the base language model. We use a publicly available checkpoint[5] from Huggingface, and fine-tune it with our dynamic graph knowledge aggregation framework. The random seed is fixed to 42 for ease of reproducibility. Our language model has 12 attention heads and 6 hidden layers in each encoder and decoder, leading to a total of 157M parameters. The maximum sequence length is limited to 512; the *batch size* is set to 64; and the *learning rate* is 1e-4. We use Adam (Kingma and Ba, 2014) as the optimiser and set its parameter to 1e-8. The whole training process lasts for 5 *epochs*. We train on an Nvidia RTX A100 GPU node, which has 120GB of system memory and 80GB of VRAM, and takes two days to train.

[2]We follow prior work (Zhang et al., 2020) in using the original validation dataset as the test set for the convenience of comparison.

[3]To our knowledge, ConceptFlow is the SOTA model for this task (where text and knowledge graphs are used as the input). There are some other similar works (Yu et al., 2022; Wu et al., 2022) in commonsense dialogue generation, but they generate dialogues with additional documents or other kind of inputs. Due to the different input formats, they cannot be considered as baselines in our task.

[4]https://pytorch.org/

[5]https://huggingface.co/thu-coai/LongLM-base

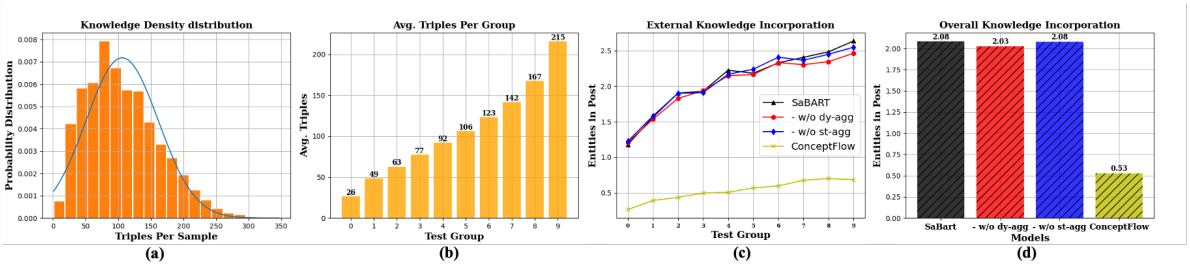

Figure 3: The analysis of knowledge incorporation. The experiment groups datasets by the amount of external knowledge they contain. We perform evaluation with the knowledge amount increasing, so that the curve in (c) demonstrates the efficacy and efficiency of utilising external knowledge when generating responses. (d) is the average used knowledge amount in the whole dataset.

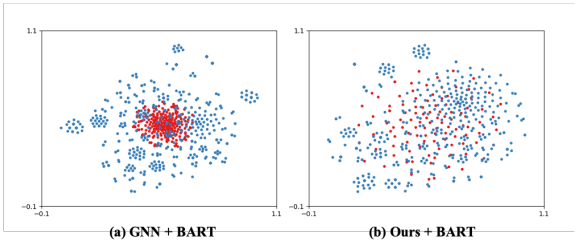

Figure 4: The embeddings of entities (in red) and generic words (in blue) contained in the post projected by t-SNE. We extract these embeddings from 30 conversational pairs.

4.3 Automatic Evaluation

Table 2 shows reference-based automatic evaluation results, which demonstrate our proposed framework substantially outperforms all baselines on all referenced metrics. In comparison to the SOTA model ConceptFlow, our model doubles (e.g. BLEU-2, NIST-2, METEOR) or triples (e.g. BLEU-3, BLEU-4) performance on most metrics. Considering the performance of - *w/o kg* (equivalent to vanilla BART), it can be inferred that the enhanced performance of our model is primarily attributable to incorporating external knowledge. Comparing to other GNN models (e.g. ConceptFlow and CCM), our model is superior in handling the heterogeneous features from the text and graph knowledge, leading to a better capturing of the global context contained in both the post and knowledge graph. In terms of unreferenced metrics, the results in Table 3 also show that our model achieves a substantial improvement in diversity. It can be observed that our model performance on Dist-1 and Dist-2 are more than twice that of the SOTA model ConceptFlow. Our improvement on both unreferenced and referenced metrics further demonstrates that the gain comes from incorporating knowledge to generate human-like responses, rather than metric-oriented training (i.e., no metric-oriented reward is used here). In addition, the ablation results of - *w/o dy-agg* and - *w/o st-agg* also prove the hierarchical layers of graph knowledge aggregation benefit the semantic understanding of graph knowledge. The aggregation of static graph features forms the representation learning of lower-level semantics, whilst the dynamic aggregation contributes to the representation of higher-level semantics. Therefore, combining the two kinds of semantics leads to a substantial performance improvement on both referenced and unreferenced metrics.

4.4 In-Depth Analysis

Furthermore, we present two experiments to analyse whether the external knowledge is better exploited in our framework than the SOTA model (ConceptFlow), as well as why our framework learns representations more efficiently and effectively.

Performance of Knowledge Incorporation. The experimental results concerning the amount of knowledge used to generate responses are illustrated in Figure 3. Firstly, we group the test set by the number of post entities retrieved in the given post. The target is to analyse the robustness and efficiency of models as the knowledge content increases. Figure 3(a) indicates the probability distribution of the knowledge amount in each conversation pair, 3(b) shows the statistics of the grouped test set, 3(c) gives the curve illustrating how many retrieved knowledge items are finally used in generated responses, and 3(d) indicates that our model substantially outperforms the SOTA model by a large margin with respect to knowledge incorporation on the whole test set. It can be ob-

served that with the proposed dynamic knowledge aggregation framework, the model tends to use more retrieved entities when generating a response. As the number of retrieved entities increases, the curve in (c) maintains a steady slope to incorporate entities, indicating that our model maintains the incorporation efficacy even with large amounts of knowledge as input. We argue that the robustness and efficiency of knowledge incorporation result from the globally aggregated features from the dynamically constructed graph with pseudo nodes, which avoids the information loss that the vanilla GNN models typically suffer. This also leads to our model outperforming other baseline models in generating high-quality responses.

Representations of Text and Graph Knowledge. Figure 4 shows the representations of text and entities from the knowledge graph. We project the embeddings of the vanilla GNN models used in the baselines into two-dimensional points for visualisation. To compare the difference in embeddings from text and the knowledge graph, we normalise by mapping both embeddings to the range of $[0, 1]$. It can be observed that the entity embeddings of the baselines (shown in Figure 4(a)) are concentrated in a circle no matter what post is given (i.e. blue points). This suggests that the GNN-learned embeddings present a biased representation for external knowledge, which may lead to difficulty in incorporating graph knowledge with the language model (which is trained on text corpora). In comparison, our framework unifies the representation hidden space of both text and graph knowledge, which makes the heterogeneous features have more shared space to fit the dataset. This mechanism makes the entity embeddings in our framework evenly spread among text words, thus it can be easily exploited by neural networks.

4.5 Human Evaluation

We present manual pair-wise comparisons to examine the *appropriateness* (whether the response is appropriate in the context) and *informativeness* (whether the response contains much information) of the most competitive baseline (ConceptFlow), our model (SaBART), as well as two ablation models (- *w/o dy-agg* and - *w/o st-agg*). Three human evaluators are instructed to give their preferred response on 100 randomly sampled conversational pairs between each compared model. The results are reported in Table 4.

Choice %	SaBART *vs* SaBART$_{\text{w/o st-agg}}$		
	SaBART	***- w/o st-agg***	*Kappa*
App.	**62.3**	37.7	0.363
Inf.	**60.0**	40.0	0.418
Choice %	**SaBART *vs* SaBART$_{\text{w/o dy-agg}}$**		
	SaBART	***- w/o dy-agg***	*Kappa*
App.	**61.9**	38.1	0.356
Inf.	**65.0**	35.0	0.438
Choice %	**SaBART *vs* ConceptFlow**		
	SaBART	**ConceptFlow**	*Kappa*
App.	**66.1**	33.9	0.423
Inf.	**70.2**	29.8	0.484

Table 4: Human Evaluation w.r.t. *appropriateness* and *informativeness*. The score is the percentage that the proposed model wins against its competitor. *Kappa* denotes Fleiss' Kappa (Fleiss, 1971), which indicates all of our evaluation annotations reach a fair or moderate agreement (meaning the two ablated models generate similar answers to **SaBART**).

When summarising the human annotation results, the final results are counted by majority voting. The ablated static aggregation and dynamic aggregation play different roles in feature incorporation, so the results of the corresponding ablation models are slightly lower than that of SaBART. On the other hand, the comparison with ConceptFlow demonstrates that our proposed model significantly outperforms the SOTA in terms of both appropriateness and informativeness, which is consistent with our observations in automatic evaluation.

4.6 Case Study

In Table 5, we show the generated responses of the most competitive models (i.e., two ablated models and ConceptFlow). We select a short post and a long post that includes more graph knowledge to validate performance. It can be observed that in both cases, the models containing aggregation frameworks tend to compose their responses with more unique and relevant entities. When a short post is given, the knowledge can effectively help avoid generating overly simplistic utterances, resulting in a more appropriate, informative, and engaging response output. Given a long input, all models seem good at generating a long response. However, compared to SaBART, the responses generated by the baseline models are less expressive due to the sub-optimal incorporation of graph knowledge. For example, ConceptFlow uses four instances of "play" in the response, diluting the

Post #1	As an energy **drink consumer** who **hates tea**, this is **hilarious**.
SaBART	I'm not a **tea drink consumer**, but i don't think it 's **hilarious**.
w/o st-agg	As a **consumer**, this is **hilarious**.
w/o dy-agg	As a **consumer**, this is **hilarious**.
ConceptFlow	You 're a **fan** of **tea** !
Post #2	**imo** the **game** deserves to be **played** on **pc**. Its not really suited for **controller**. I **played** it on **ps3** and sometimes it was just **frustrating**.
SaBART	I've **played** it on **ps3**, and i have no **idea** what the **problem** is with it. **pc** is really **suitable** for this **game**.
w/o st-agg	It 's not suited for **pc**, but it doesn't have to be on a **ps3** .
w/o dy-agg	I've **played** it on **ps3**, and it's not really suited for **pc**.
ConceptFlow	I **played** it for about a **year** and then i got to **play** it on my **ps3** and it was the only **game** i ever **played**. I didn't even know how to **play** it .

Table 5: The case study for two conversational pairs extracted from the test dataset. Words in **bold** denote the graph entities in the posts (entity mentions) or in the responses (used knowledge).

information it conveys, in addition to the content not being coherent or related to the post. In contrast, SaBART is able to better exploit the retrieved knowledge (e.g. "frustrating" in relation to "problem"), which thus results in composing a more appropriate and informative response.

5 Conclusion

In this study, we propose a novel dynamic graph aggregation framework for the task of knowledge graph enhanced dialogue generation. We dynamically construct a graph with created pseudo nodes, and hierarchically aggregate graph knowledge to attain a global feature which better represents the topological structure and semantics contained in the knowledge graph. Due to the superior ability in leveraging the heterogeneous features of input text and graph knowledge, our framework can fill the semantic gap between the language model and knowledge graph and consequently generate an informative response. The extensive experiments demonstrate that our model significantly outperforms all baseline models, and can generate more appropriate and informative responses utilising external graph knowledge.

Limitations

This paper aims to investigate a more efficient and effective framework to incorporate the heterogeneous features of both text and graph knowledge. The extensive experiments demonstrate our framework has a superior performance in capturing semantics of input knowledge, thus beating all SOTA models. However, due to the time and resource limit, we could not conduct further experimentation to compare with promising frameworks in similar areas. In fact, we have observed some other techniques (Tang et al., 2022c; Yu et al., 2022; Wu et al., 2022) may be beneficial to our study, but when considering the difficulty in applying them here (due to additional annotation and knowledge being required), we have to leave them to future work. We also cannot exclude some other factors which may affect performance. For example, we select BART as the base language model in this paper. In practical use, the latest language models (e.g. ChatGPT) may have better performance in this task. We have to leave the analysis of these factors to future study.

Ethics Statement

We conduct the experiments based on an existing publicly available dataset from Zhou et al. (2018) which is a large-scale dataset widely used to study commonsense dialogue generation, and we strictly follow the license and instructions. We also read and acknowledge the ACM Code of Ethnics and Professional Conduct.[6] We take our professional responsibilities very seriously, and our study did not violate any ethical principles. Additionally, whilst our work concerns the incorporation of knowledge from knowledge graphs in dialogue systems, we acknowledge that the veracity and validity of the knowledge in such resources should be assessed in production, in order to avoid the perpetuation of misinformation.

Acknowledgements

Chen Tang is supported by the China Scholarship Council (CSC) for his doctoral study (File No.202006120039). Tyler Loakman is supported

[6] https://www.acm.org/code-of-ethics

by the Centre for Doctoral Training in Speech and Language Technologies (SLT) and their Applications funded by UK Research and Innovation [grant number EP/S023062/1].

References

Shaked Brody, Uri Alon, and Eran Yahav. 2021. How attentive are graph attention networks? *arXiv preprint arXiv:2105.14491*.

George Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the Second International Conference on Human Language Technology Research*, HLT '02, page 138–145, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. *Advances in Neural Information Processing Systems*, 32.

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.

Marjan Ghazvininejad, Chris Brockett, Ming-Wei Chang, Bill Dolan, Jianfeng Gao, Wen-tau Yih, and Michel Galley. 2018. A knowledge-grounded neural conversation model. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

Henglin Huang, Chen Tang, Tyler Loakman, Frank Guerin, and Chenghua Lin. 2022. Improving Chinese story generation via awareness of syntactic dependencies and semantics. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, Online only.

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

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Alon Lavie and Abhaya Agarwal. 2007. METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 228–231, Prague, Czech Republic. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online.

Jinjie Ni, Tom Young, Vlad Pandelea, Fuzhao Xue, and Erik Cambria. 2022. Recent advances in deep learning based dialogue systems: A systematic survey. *Artificial intelligence review*, pages 1–101.

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

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

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

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27.

Chen Tang, Weile Chen, Tao Wang, Chun Sun, JingChi Jiang, and Yi Guan. 2021. Normcg: A novel deep learning model for medical entity linking. In *Intelligent Data Engineering and Analytics*, pages 565–573. Springer.

Chen Tang, Chenghua Lin, Henglin Huang, Frank Guerin, and Zhihao Zhang. 2022a. Etrica: Event-triggered context-aware story generation augmented by cross attention. *arXiv preprint arXiv:2210.12463*.

Chen Tang, Hongbo Zhang, Tyler Loakman, Chenghua Lin, and Frank Guerin. 2022b. Terminology-aware medical dialogue generation. *arXiv preprint arXiv:2210.15551*.

Chen Tang, Zhihao Zhang, Tyler Loakman, Chenghua Lin, and Frank Guerin. 2022c. NGEP: A graph-based event planning framework for story generation. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, Online only. Association for Computational Linguistics.

Yi-Lin Tuan, Yun-Nung Chen, and Hung-yi Lee. 2019. Dykgchat: Benchmarking dialogue generation grounding on dynamic knowledge graphs. *arXiv preprint arXiv:1910.00610*.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *stat*, 1050:20.

Sixing Wu, Minghui Wang, Ying Li, Dawei Zhang, and Zhonghai Wu. 2022. Improving the applicability of knowledge-enhanced dialogue generation systems by using heterogeneous knowledge from multiple sources. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, WSDM '22, page 1149–1157, New York, NY, USA. Association for Computing Machinery.

Wenhao Yu, Chenguang Zhu, Lianhui Qin, Zhihan Zhang, Tong Zhao, and Meng Jiang. 2022. Diversifying content generation for commonsense reasoning with mixture of knowledge graph experts. In *Proceedings of the 2nd Workshop on Deep Learning on Graphs for Natural Language Processing (DLG4NLP 2022)*, pages 1–11, Seattle, Washington. Association for Computational Linguistics.

Houyu Zhang, Zhenghao Liu, Chenyan Xiong, and Zhiyuan Liu. 2020. Grounded conversation generation as guided traverses in commonsense knowledge graphs. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Yizhe Zhang, Michel Galley, Jianfeng Gao, Zhe Gan, Xiujun Li, Chris Brockett, and Bill Dolan. 2018. Generating informative and diverse conversational responses via adversarial information maximization. *Advances in Neural Information Processing Systems*, 31.

Xueliang Zhao, Wei Wu, Can Xu, Chongyang Tao, Dongyan Zhao, and Rui Yan. 2020. Knowledge-grounded dialogue generation with pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online.

Hao Zhou, Tom Young, Minlie Huang, Haizhou Zhao, Jingfang Xu, and Xiaoyan Zhu. 2018. Commonsense knowledge aware conversation generation with graph attention. In *IJCAI*, pages 4623–4629.

Wenya Zhu, Kaixiang Mo, Yu Zhang, Zhangbin Zhu, Xuezheng Peng, and Qiang Yang. 2017. Flexible end-to-end dialogue system for knowledge grounded conversation. *arXiv preprint arXiv:1709.04264*.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Line 547 to 567

☑ A2. Did you discuss any potential risks of your work?
Line 568 to 582

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract (all) and Introduction (Lines 082 - 145)

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

The commonsense dialogue generation dataset from https://github.com/thu-coai/ccm .

☑ B1. Did you cite the creators of artifacts you used?
Sec. 4.1

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Section of "Ethnical Statement"

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Section of "Ethnical Statement"

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Not applicable. Left blank.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Section 4.1

C ☑ Did you run computational experiments?

Section 4

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section 4.2

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 4.2

☒ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
a single run

☒ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
I use ROUGE from a python package, and it needs no parameter settings.

D ☑ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Section 4.5

☑ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Section 4.5

☑ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
students

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Not applicable. Left blank.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. Left blank.