applied sciences

MDPI

*Article*

# Generative Model Using Knowledge Graph for Document-Grounded Conversations

**Boeun Kim [1], Dohaeng Lee [1], Damrin Kim [1], Hongjin Kim [1], Sihyung Kim [2,*], Oh-Woog Kwon [3] and Harksoo Kim [4,*]**

[1] Artificial Intelligence, Konkuk University, Seoul 05029, Korea; boeun@konkuk.ac.kr (B.K.); dsdhlee@konkuk.ac.kr (D.L.); ekafls33@konkuk.ac.kr (D.K.); jin3430@konkuk.ac.kr (H.K.)

[2] NAVER Corporation, Seongnam 13561, Korea

[3] Language Intelligence Research Lab., Electronics and Telecommunications Research Institute, Daejeon 34129, Korea; ohwoog@etri.re.kr

[4] Computer Science and Engineering, Konkuk University, Seoul 05029, Korea

[*] Correspondence: sihyung.kim@navercorp.com (S.K.); nlpdrkim@konkuk.ac.kr (H.K.); / Tel.: +82-2-450-3499 (H.K.)

**Featured Application: Core technology for document-grounded conversation.**

**Abstract:** Document-grounded conversation (DGC) is a natural language generation task to generate fluent and informative responses by leveraging dialogue history and document(s). Recently, DGCs have focused on fine-tuning using pretrained language models. However, these approaches have a problem in that they must leverage the background knowledge under capacity constraints. For example, the maximum length of the input is limited to 512 or 1024 tokens. This problem is fatal in DGC because most documents are longer than the maximum input length. To address this problem, we propose a document-grounded generative model using a knowledge graph. The proposed model converts knowledge sentences extracted from the given document(s) into knowledge graphs and fine-tunes the pretrained model using the graph. We validated the effectiveness of the proposed model using a comparative experiment on the well-known Wizard-of-Wikipedia dataset. The proposed model outperformed the previous state-of-the-art model in our experiments on the Doc2dial dataset.

**Keywords:** document-grounded conversation; generative model; knowledge graph; Levi graph; open information extraction

**Citation:** Kim, B.; Lee, D.; Kim, D.; Kim, H.; Kim, S.; Kim, O.-W.; Kim, H. Generative Model Using Knowledge Graph for Document-Grounded Conversations. *Appl. Sci.* **2022**, *12*, 3367. https://doi.org/10.3390/app12073367

Academic Editor: Mauro Castelli and Luca Manzoni

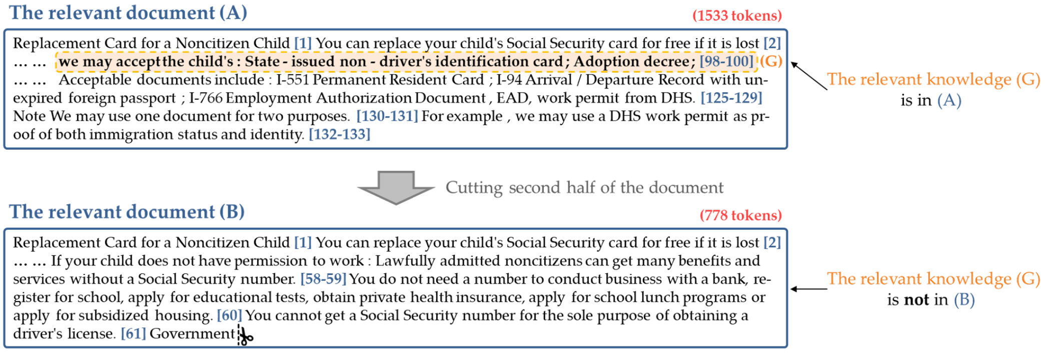
Received: 21 February 2022
Accepted: 24 March 2022
Published: 25 March 2022

## 1. Introduction

One of the various goals of the natural language generation (NLG) tasks is to build generative models that can think and talk like a human being. Document-grounded conversation (DGC), a task in NLG, generates fluent and informative responses similar to a human being by leveraging diverse contexts (e.g., dialogue history and document(s)). The document-grounded generative model (DGGM) focuses on topical conversations about a specific topic, since humans not only have small conversations based on common sense but also have topical conversations associated with books, documents, and articles. Sometimes, humans have a conversation using the knowledge of a specific topic. Thus, the DGGM should be able to extract the relevant knowledge from dialogue history or relevant document(s), refine it, and utilize the knowledge for response generation. Furthermore, the DGGM could leverage external knowledge accumulated by humans, such as a Wikipedia dump. In this case, the DGGM can generate more informative responses using profound knowledge when discussing topics in depth. Hence, the applicability of external knowledge is significant for DGGMs.

Recently, with the rapid development of pretraining approaches, many researchers have proposed the transformer-based methods to apply a pretrained language model (PLM) such as GPT-2 [1] and BART [2] to the document-grounded conversation (DGC) [3–5].

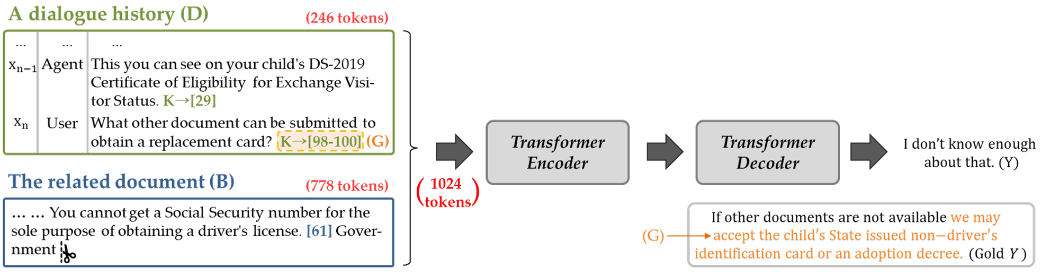
However, PLMs suffer from capacity constraints. PLMs input a given source data (e.g., dialogue history, the relevant document(s)) within the maximum length of the input. According to the model size, PLMs are set to the maximum length of the input differently (e.g., a small-sized model, base-sized model, and large-sized model set 256 tokens, 512 tokens, and 1024 tokens, respectively). In other words, PLM-based DGGM takes the part of the document(s) as an input under capacity constraints, as shown in Figures 1 and 2. As a result, PLM-based DGGM could miss important information of input texts. Therefore, this problem leads PLM-based DGGM to suffer from generating the proper responses. To solve this problem, we assume that key contents in the document(s) can be carried through knowledge graphs. Based on this assumption, we leverage knowledge graphs that can sum up the document(s), as shown in Figure 3. We expect that the knowledge graphs contain the crucial information of the document regardless of its length.


The relevant document (A)
(1533 tokens)
The relevant knowledge (G)
is in (A)
Cutting second half of the document
The relevant document (B)
(778 tokens)
The relevant knowledge (G)
is **not** in (B)

Replacement Card for a Noncitizen Child [1] You can replace your child's Social Security card for free if it is lost [2] … … **we may accept the child's : State - issued non - driver's identification card ; Adoption decree ;** [98-100] (G) … … Acceptable documents include : I-551 Permanent Resident Card ; I-94 Arrival / Departure Record with unexpired foreign passport ; I-766 Employment Authorization Document , EAD, work permit from DHS. [125-129] Note We may use one document for two purposes. [130-131] For example , we may use a DHS work permit as proof of both immigration status and identity. [132-133]

Replacement Card for a Noncitizen Child [1] You can replace your child's Social Security card for free if it is lost [2] … … If your child does not have permission to work : Lawfully admitted noncitizens can get many benefits and services without a Social Security number. [58-59] You do not need a number to conduct business with a bank, register for school, apply for educational tests, obtain private health insurance, apply for school lunch programs or apply for subsidized housing. [60] You cannot get a Social Security number for the sole purpose of obtaining a driver's license. [61] Government

**Figure 1.** Process of truncating the relevant document.



| | | |
|---|---|---|
| … | … | … |
| $x_{n-1}$ | Agent | This you can see on your child's DS-2019 Certificate of Eligibility for Exchange Visitor Status. K→[29] |
| $x_n$ | User | What other document can be submitted to obtain a replacement card? K→[98-100] (G) |

… … You cannot get a Social Security number for the sole purpose of obtaining a driver's license. [61] Government

I don't know enough about that. (Y)

If other documents are not available we may accept the child's State issued non−driver's identification card or an adoption decree. (Gold *Y* )

**Figure 2.** Capacity constraint problem when the model uses a long knowledge document.

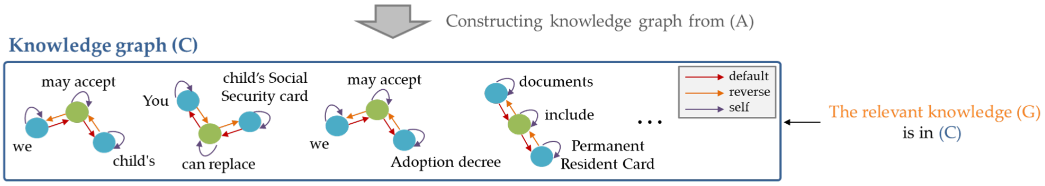


**Figure 3.** Process of constructing knowledge graph from the relevant document (A).

The relevant document (A) (which has 1533 tokens) and a dialogue history $X = \{x_1, \ldots, x_{n-1}, x_n\}$ (which has 246 tokens) (D) are used to train the pretrained model, as shown in Figures 1 and 2. However, because of constraints on the maximum number of tokens (e.g., the maximum number for BART is 1024), most of the PLMs often cannot use the entire source (A, D) [6]. To fit the maximum length of the input, we truncate the length of the document (A) by cutting the second half of it, that is, (B) is 50.7% of (A). Then, given a dialogue history (D) and the first half of the document (B), the model generates an informative and natural agent response $Y$. To generate an informative response $Y$, the model needs gold relevant knowledge (G), $K \rightarrow [98, 100]$, which is related to the previous turn, $x_n$ knowledge (G) is present in the original document (A), not in the truncated document (B). If the relevant document is too long, the relevant knowledge (G) may not be in the input, according to the location of the knowledge in the document, such as in Figure 2. The relevant knowledge (G) is not present in the input, which is a serious problem in DGC. To address this problem, we propose a document-grounded generative model using a knowledge graph. Figure 3 shows the construction of a knowledge graph (C). We convert the sentences of document (A) into the knowledge graph (C) with knowledge triples using keywords of the sentences. The proposed model dynamically constructs a knowledge graph from relevant document(s), and generates more informative and natural responses based on the graphs. The remainder of this paper is organized as follows. In Section 2, we review previous studies on DGC. In Section 3, we describe the proposed model, a document-grounded generative model using knowledge graph. In Section 4, we explain our experiments and report the experimental results. In Section 5, we provide the conclusion of our work.

## 2. Previous Studies

Many studies on DGC are based on sequence-to-sequence (Seq2Seq) modeling [7]. Most early studies used recurrent neural networks (RNNs) [8–10]. In these studies, an encoder–decoder model utilizing RNN was used. These generative models based on RNN encode an extensive knowledge base using attention mechanisms [11] or memory networks [12], and then generate responses by adding that knowledge context to the dialogue context. However, this approach suffers from performance degradation when extended contexts are used. In addition, the training time is excessively long. Many studies have recently used pretrained transformer-based language models [4,5], such as BERT [6]. These models are trained on a large-scale corpus, and they can be fine-tuned using a task-specific dataset. In DGC, GPT-2 [1], or BART [2], a pretrained language model is mainly used.

He et al. [13] concatenated the knowledge context with the dialogue context and encoded them using self-attention to provide background knowledge to the model. However, this mechanism suffers from capacity constraints. For example, the entire contexts cannot be used if the dialogue context or knowledge context is too long. To solve this problem, several studies used machine reading comprehension (MRC) [14] and a knowledge selector (KS) [15]. These studies extracted context-relevant knowledge or snippets from relevant document(s) and generated responses using them as inputs. However, these studies have a problem in that the generative model is highly dependent on the performance of the MRC or KS.

## 3. Document-Grounded Generative Model Using Knowledge Graph

The proposed model, which is a document-grounded generative model using knowledge graph networks, comprises two steps: constructing knowledge graph networks and training the model (see Figure 4).

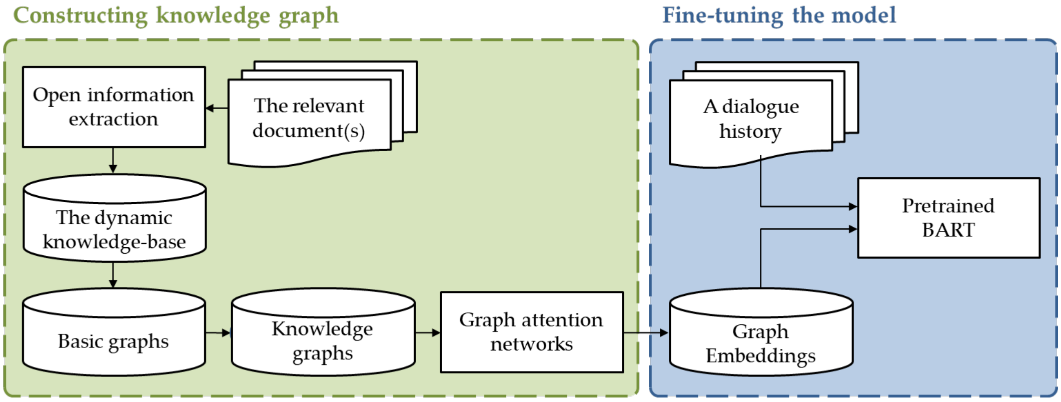


**Figure 4.** The overall workflow of the proposed model.

The first step is extracting knowledge triples using the open information extraction model (OpenIE) [16] from the relevant document(s), integrating the triples into basic graphs, and converting the graphs into Levi graphs [17]. The second step is to choose a PLM and fine-tune it. We adopted BART with six encoder and decoder layers, a base-sized model, [2] as a pretrained language model and implemented the model using transformers (https://huggingface.co/transformers/, accessed on 25 May 2022). Furthermore, we fine-tuned the model using a knowledge graph.

### 3.1. Constructing a Knowledge Graph

OpenIE can recognize the entities, nouns that have unique meaning (e.g., proper nouns), in a sentence and convert their relations to literary triples. Hence, the name of the relation of the triple may include part of the words of the entity. We extract knowledge triples (i.e., subject, relation, object) using OpenIE from the relevant document(s) to build the dynamic knowledge base. We can flexibly extract the triples from open-domain document(s) because OpenIE is available when relation type labels are absent. If the boundary of the entity is blurred, the boundary is expanded by the window size. The maximum window size is 5. For example, given a window size of one and the location of the center word (t), the location of the same entity is (t − 1), (t), (t + 1). In other words, to alleviate the ambiguity of the boundary, we expand the boundary next to the center word so that it belongs to the same entity. Table 1 presents a part of the triples extracted from the sentence. As shown in Table 1, the triples extracted from the sentence "Obama was the first African-American president of the United States" are as follows: "(Obama; president of; United States)", "(Obama; was; African-American president)", and "(Obama; American president of; United States)". As shown in Table 1, the words "American president" can be included in "relation" or "object".

**Table 1.** Sample of knowledge triples extracted from the sentence using OpenIE.

| **Sentence** | Obama Was the first African-American President of the United States. | | |
|---|---|---|---|
| **Knowledge triples** | **Subject** | **Relation** | **Object** |
| | Obama | president of | United States |
| | Obama | was | African-American president |
| | Obama | American president of | United States |

Subsequently, to use the triples as input, we integrate the triples into basic graphs and convert them into knowledge graphs using a Levi graph [17]. In the phase of integrating the triples, we create nodes based on the "subject" and "object" of the triple and an edge based on the "relation" of the triple. We then connect the two nodes via the edge. In the graph conversion phase, we convert the edge into a node and connect the three nodes via new edges as follows: "default", "reverse", and "self". "Default" is an edge that points in the existing direction. "Reverse" is an edge that points in the reverse direction of the edge "default". "Self" is an edge that points to itself. Figure 5 shows the conversion of a basic graph into a knowledge graph. As shown in Figure 5, in a basic graph, "Obama" (subject) and "United States" (object) of the triple are nodes, and "president of" (relation) of the triple is an edge. In a knowledge graph, "president of" (relation) is converted into a node.

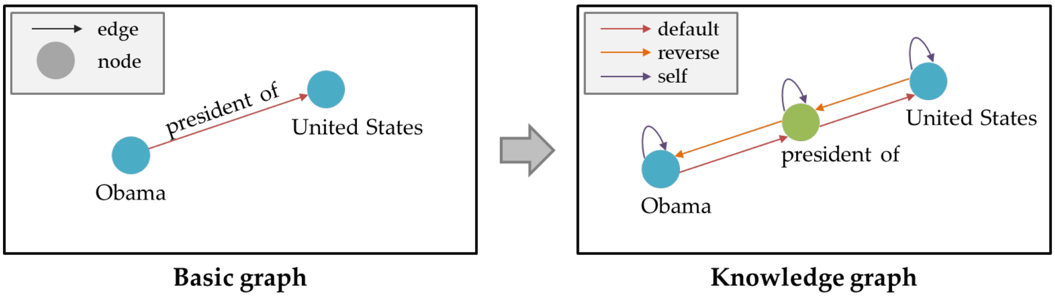


**Figure 5.** Example of converting a basic graph into a knowledge graph.

### 3.2. Fine-Tuning the Model

We adopt BART as a pretrained language model to fine-tune the proposed model using knowledge graphs with a dialogue history. The proposed model is a transformer-based encoder–decoder model. Figure 6 illustrates the architecture of the proposed model.

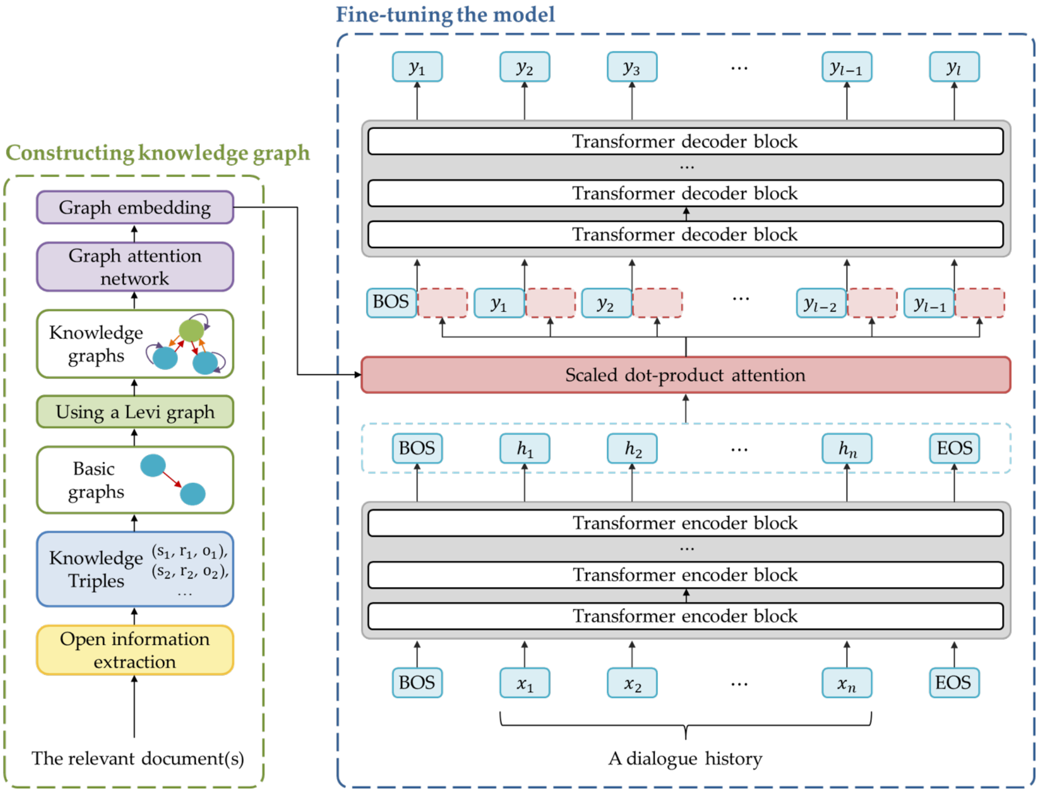

Fine-tuning the model
Constructing knowledge graph
Graph embedding
Graph attention network
Knowledge graphs
Using a Levi graph
Basic graphs
Knowledge Triples
(s1, r1, o1), (s2, r2, o2), …
Open information extraction
The relevant document(s)
y1 y2 y3 … yl−1 yl
Transformer decoder block
…
Transformer decoder block
Transformer decoder block
BOS y1 y2 … yl−2 yl−1
Scaled dot-product attention
BOS h1 h2 … hn EOS
Transformer encoder block
…
Transformer encoder block
Transformer encoder block
BOS x1 x2 … xn EOS
A dialogue history

**Figure 6.** The overall architecture of the proposed model. The model comprises two parts; constructing knowledge graph and fine-tuning the model. The red dotted blocks indicate the attention scores that are calculated by using a scaled-dot product.

In the encoder layer, given the dialogue contexts $X = \{x_1, x_2, \ldots, x_n\}$, we concatenate $BOS$, $X$, and $EOS$ tokens. We then use a transformer-based BART encoder to convert the contexts into hidden states $H = \{h_1, h_2, \ldots, h_n\}$, as shown in the following equation:

$$H = BARTEncoder(BOS; X; EOS) \tag{1}$$

where $BOS$ denotes the token to present the beginning of a sequence, and $EOS$ denotes the token to present the end of a sequence. The knowledge graphs constructed as described in

Section 3.1 are encoded using a graph attention network (GAT) [18] to leverage the relevant document(s), as shown in the following equation:

$$\begin{aligned} \vec{g}_i' &= \sigma\left(\frac{1}{K}\sum_{k=1}^{k}\sum_{j\in N_i}\alpha_{ij}^{k}W^{k}\vec{g}_j\right), \\ G &= \left[\vec{g}_1', \vec{g}_2', \ldots, \vec{g}_N'\right] \end{aligned} \tag{2}$$

where $\vec{g}_i'$ denotes the final output representation of the $i$-th node ($\vec{g}_i' \in \mathbb{R}^{N\times d_k}$, where $d_k$ denotes the BART embedding size), $K$ is the number of heads, $W^k$ is the corresponding input linear transformation's weight matrix, $N_i$ are neighbor nodes of the $i$-th node in the graph, and $\alpha_{ij}^k$ is normalized attention coefficients). In the attention layer, matrices $Attention(H, G, G)$ to the hidden states $H$ use a residual connection [19] to obtain matrix $F$, which reflects background knowledge, as shown in the following equation:

$$\begin{aligned} Attention(H, G, G) &= softmax\left(\frac{HG^T}{\sqrt{d_k}}\right)G, \\ F &= Attention(H, G, G) + H \end{aligned} \tag{3}$$

where $Attention(H, G, G)$ is calculated using a scaled dot-product attention mechanism [11], and $d_k$ is a normalization factor. In the decoder layer, matrix $F$ is used as the input for a transformer-based BART decoder to generate the token $y_i$.

$$y_i = BARTDecoder(F, y_{i-1}) \tag{4}$$

where $y_i$ denotes the token generated at each decoding step, and $y_0$ is the $BOS$ token that is input into the decoder. Then, $y_{i-1}$ is input differently into the decoder according to the training and inference phases. Because the BART decoder is an autoregressive model, the gold tokens are input in the training phase to generate the following tokens. The previously generated tokens are input in the inference phase.

## 4. Evaluation

### *4.1. Datasets and Experimental Settings*

We evaluated the proposed model using the Wizard-of-Wikipedia (WoW) dataset [5] and the Doc2dial dataset [14]. The WoW dataset is an open-domain dialogue dataset. It contains multi-turn conversations between two speakers, an apprentice, and a wizard. The relevant documents are not accessible to the apprentice, whereas the wizard can access knowledge from relevant documents during chatting. The apprentice plays the role of a curious learner about a chosen document (topic). The wizard plays the role of a knowledgeable expert, keeping the conversation engaging. The Doc2dial dataset is a goal-oriented dialogue dataset. It contains multi-turn conversations grounded in relevant documents from four domains for social welfare: SSA, VA, DMV, and student aid. The conversations of Doc2dial have two speakers; a user and an agent. The two speakers aim to generate responses and inquiries with document-based or dialogue-based contexts, maintaining a free-form conversation. As shown in Table 2, in the WoW dataset, we removed outliers with more than 10,000 tokens. The average length of the documents is 2644.5 tokens, and the maximum length is 9935 tokens. The documents of the WoW dataset refer to the top passages retrieved for the last two turns of dialogue (by the apprentice and wizard) and the present turn, containing the gold knowledge. In the Doc2dial dataset, the average length of the documents is 964.7 tokens, and the maximum length is 8891 tokens. The documents of the Doc2dial dataset represent the given documents for the conversation. These datasets are suitable for our experiments because most documents are longer than the maximum input length.

**Table 2.** Statistics of the document length for both the Wizard-of-Wikipedia dataset and the Doc2dial dataset.

| Dataset | Minimum | Maximum | Average |
|---|---|---|---|
| WoW | 592 | 9935 | 2644.5 |
| Doc2Dial | 174 | 8891 | 964.7 |

We evaluated our model using the following evaluation measures: F1-score [5,9,15], BLEU [20,21], and SacreBLEU [22]. We used the F1-score and BLEU for the WoW dataset. BLEU measures the number of generated responses that contain word n-gram overlaps with the gold responses, as shown in the following equation:

$$BLEU = min\left(1, \frac{Length\ of\ a\ generated\ sentence}{Length\ of\ a\ gold\ sentence}\right)\left(\prod_{i=1}^{n} precision_i\right)^{\frac{1}{n}} \tag{5}$$

where $n$ denotes the maximum length of $n$-grams, which is commonly set to 4, and $precision_i$ is the word $i$-gram precision (i.e., the number of correct word $i$-grams divided by the number of word $i$-grams in a generated sentence). F1-score is calculated from precision and recall, as shown in the following equation:

$$\begin{aligned} precision &= \frac{W_{gold} \cap W_{generated}}{|W_{generated}|} \\ recall &= \frac{W_{gold} \cap W_{generated}}{|W_{gold}|} \\ F1 - score &= \frac{2 \times precision \times recall}{precision + recall} \end{aligned} \tag{6}$$

where $W_{gold}$ denotes a unigram word list of the gold response, $W_{generated}$ denotes a unigram word list of the generated response, and $|W|$ indicates the length of the word list. We used SacreBLEU for the Doc2dial dataset because it was used to evaluate DialDoc Shared Task Subtask2.

### 4.2. Experimental Results

In the first experiment, we measured the rate of knowledge loss due to capacity constraints. We measured whether the document had the necessary knowledge (i.e., the gold knowledge) to generate responses when the document was truncated to the maximum length for BART (1024 tokens).

Table 3 shows the rate of gold knowledge in the truncated documents. As shown in Table 3, for the WoW dataset, the percentage of truncated documents containing gold knowledge was 49.39%, whereas it was 85.66% for the Doc2dial dataset. In other words, some gold knowledge required for responses was unavailable for both datasets. In particular, over half of the truncated documents (50.61%) did not contain the gold knowledge for the WoW dataset.

**Table 3.** Percentage of the truncated documents containing gold knowledge both for the Wizard-of-Wikipedia dataset and the Doc2dial dataset.

| Dataset | Percentage of the Documents |
|---|---|
| Wizard-of-Wikipedia | 49.39% |
| Doc2Dial | 85.66% |

In the second experiment, we measured the effectiveness of converting the document into a graph on the Wizard-of-Wikipedia dataset. The results are listed in Table 4.

**Table 4.** Performance comparison according to knowledge input on the Wizard-of-Wikipedia dataset.

| Document Knowledge | BLEU-1 | BLEU-2 | BLEU-3 | BLEU-4 | F1-Score |
|---|---|---|---|---|---|
| Gold knowledge | 0.282 | 0.207 | 0.165 | 0.137 | 0.378 |
| Document (100%) | 0.134 | 0.065 | 0.038 | 0.025 | 0.182 |
| Document (75%) | 0.132 | 0.064 | 0.037 | 0.024 | 0.183 |
| Document (50%) | 0.133 | 0.065 | 0.038 | 0.025 | 0.183 |
| Document (25%) | 0.130 | 0.064 | 0.038 | 0.025 | 0.179 |
| Document (0%) | 0.107 | 0.049 | 0.025 | 0.015 | 0.171 |
| Knowledge graph (our model) | 0.152 | 0.074 | 0.042 | 0.026 | 0.193 |

In Table 4, gold knowledge indicates that the gold knowledge was input as the knowledge context into the BART model. Document ($k$%) indicates the truncated document(s) input into the model, and $k$ denotes the first half $k$% of the original document(s). The knowledge graph (our model) constructed as described in Section 3.1 was also input into the model. As shown in Table 4, the document(s) input into the model show inferior performance. Even if the entire document is input, there is a performance limitation. However, our model outperformed in terms of all measures. This implies that the graph approach effectively condensed the documents.

In the third experiment, we compared the performance of the proposed model with that of the state-of-the-art model on the Doc2dial dataset.

In Table 5, the baseline model (gold knowledge) uses BART to input the gold knowledge. The baseline model (document (100%)) uses BART to input truncated document(s). KU_NLP [23] is a transformer-based model that uses an MRC model to identify the necessary knowledge, and then takes the knowledge as input with diverse embeddings; it won the first prize at the DialDoc21: Shared Task Subtask2 on the Doc2dial dataset. As shown in Table 5, our model outperformed all the comparison models. This suggests that the graph approach is helpful in generating more informative responses than the MRC approach.

**Table 5.** Performance comparison with state-of-the-art models on the Doc2dial dataset.

| Model | SacreBLEU |
|---|---|
| Baseline model (gold knowledge) | 32.96 |
| Baseline model (document (100%)) | 21.90 |
| KU_NLP [23] | 25.28 |
| Our model | 25.71 |

In the fourth experiment, we compared the performance according to kinds of graphs on the Wizard-of-Wikipedia dataset.

In Table 6, Levi (ours) is the graph conversion method that we adopted in this paper. Graphdialogue [24] is another graph conversion method, and we simply replaced Levi (ours) into Graphdialogue for comparison. Levi (ours) performed better than Graphdialogue. This indicates that Levi (ours) is more effective than Graphdialogue in constructing knowledge graphs.

**Table 6.** Performance comparison according to kinds of graphs on the Wizard-of-Wikipedia dataset.

| Graph representation | BLEU-1 | BLEU-2 | BLEU-3 | BLEU-4 | F1-score |
|---|---|---|---|---|---|
| Graphdialog [24] | 0.149 | 0.072 | 0.041 | 0.026 | 0.186 |
| Levi (ours) | 0.152 | 0.074 | 0.042 | 0.026 | 0.193 |

Table 7 shows an example of the response of the baseline model (Document(100%)) and our model on the Wizard-of-Wikipedia dataset.

**Table 7.** An example of the response on the Wizard-of-Wikipedia dataset.

| | |
|---|---|
| The relevant document(s) | purple is a color intermediate between blue and red. it is similar to violet, but unlike violet, which is a spectral color with its own wavelength on the visible spectrum of light, purple is a composite color made by combining red and blue. according to surveys in europe and the u.s., purple is the color most often associated with royalty, magic, mystery and piety. ... |
| Knowledge graph | [[[”Pur”, “ple”], [“is”], [“color”, “inter”, “mediate”, “between”, “blue”]], [[“Pur”, “ple”], [“is”], [“color”, “inter”, “mediate”]], [[“Pur”, “ple”], [“is”], [“inter”, “mediate”]], [[“Pur”, “ple”], [“is”], [“color”]], [[“sur”, “ve”, “ys”], [“in”], [“Europe”]], [[“it”], [“is”], [“associated”]], [[“it”], [“is”, “associated”, “with”], [“er”, “otic”, “ism”]], [[“it”], [“comb”, “ined”, “with”], [“p”, “ink”]], [[“Pur”, “ple”], [“was”], [“color”, “worn”]], [[“it”, “bec”, “ame”, “the”, “im”, ”perial“, ”color”], [“worn”, “by”], [“r”, “ul”, “ers”, “of”, “the”, “By”, “z”, “antine”, “E”, “mpire”]], ... |
| A dialogue history | Oh that’s cool, do you know if purple had any historical uses? I don’t know much about the color except that it is an intermediate color between blue and red. I do like it though. Purple My favorite color is purple, do you know much about that color? |
| Ground truth | I would say that it was useful in draping the past Royalty of Europe and other countries, as well as many magician’s ensembles. |
| Baseline model (Document(100%)) | I don’t know much about that, but I do know that purple was the color worn by Roman magistrates. |
| Our model | I’m not sure but I do know that purple is associated with royalty, magic and mystery. |

The baseline model (document(100%)) generated incorrect responses to the last utterance of a dialogue history. However, our model generated informative responses containing the fact “purple is associated with royalty” by using knowledge graphs.

## 5. Conclusions

In this work, we proposed a document-grounded generative model using a knowledge graph. To input all knowledge into the model, we extracted triples from the relevant document(s) using OpenIE and converted the triples into knowledge graphs using Levi graphs. Finally, we proposed an approach to effectively connect the graphs to a pretrained model using graph attention networks. In our experiment using the WoW dataset, we demonstrated the effectiveness of the proposed model. In addition, in our experiment using the Doc2dial dataset, the proposed model outperformed other models and the state-of-the-art model. We conclude that the knowledge graphs are useful to sum up the document(s). Additionally, we conclude that the proposed model effectively alleviates capacity constraint problems of language models (e.g., BART, GPT-2). In future studies, we intend to focus on a method to construct knowledge graphs using coreference resolution to improve the performance.

**Author Contributions:** Conceptualization, H.K. (Harksoo Kim); methodology, H.K. (Harksoo Kim); software, B.K. and S.K.; validation, D.L., D.K., and H.K. (Hongjin Kim); formal analysis, H.K. (Harksoo Kim); investigation, H.K. (Harksoo Kim); resources, B.K., D.L., D.K., and H.K. (Hongjin Kim); data curation, B.K. and S.K.; writing—original draft preparation, B.K. and S.K.; writing—review and editing, H.K. (Harksoo Kim); visualization, H.K. (Harksoo Kim); supervision, H.K. (Harksoo Kim) and O.-W.K.; project administration, H.K. (Harksoo Kim) and O.-W.K.; funding acquisition, H.K. (Harksoo Kim) and O.-W.K. All authors have read and agreed to the published version of the manuscript.

**Funding:** This paper was supported by Konkuk University Researcher Fund in 2021. This work was also supported by Institute of Information & Communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (2019-0-00004, Development of semi-supervised learning language intelligence technology and Korean tutoring service for foreigners) and the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. 2020R1F1A1069737).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Publicly available datasets were analyzed in this study. This data can be found here: https://parl.ai/projects/wizard_of_wikipedia/ (accessed on 25 May 2022) and https://github.com/doc2dial/sharedtask-dialdoc2021/ (accessed on 25 May 2022).

**Acknowledgments:** We thank the members of the NLP laboratory at Konkuk University for their technical support.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language models are unsupervised multitask learners. *OpenAI Blog* **2019**, *1*, 9.
2. Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; Zettlemoyer, L. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv* **2019**, arXiv:1910.13461.
3. Zhao, X.; Wu, W.; Xu, C.; Tao, C.; Zhao, D.; Yan, R. Knowledge-grounded dialogue generation with pre-trained language models. *arXiv* **2020**, arXiv:2010.08824.
4. Bao, S.; He, H.; Wang, F.; Wu, H.; Wang, H. Plato: Pre-trained dialogue generation model with discrete latent variable. *arXiv* **2019**, arXiv:1910.07931.
5. Dinan, E.; Roller, S.; Shuster, K.; Fan, A.; Auli, M.; Weston, J. Wizard of Wikipedia: Knowledge-powered conversational agents. *arXiv* **2018**, arXiv:1811.01241.
6. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
7. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*; Montreal, Canada, 8-13 Dec 2014; Volume 27. Available online: https://proceedings.neurips.cc/paper/2014/hash/a14ac55a4f27472c5d894ec1c3c743d2-Abstract.html (accessed on 25 May 2022).
8. Kim, J.; Oh, S.; Kwon, O.W.; Kim, H. Multi-turn chatbot based on query-context attentions and dual Wasserstein generative adversarial networks. *Appl. Sci.* **2019**, *9*, 3908.
9. Kim, S.; Kwon, O.W.; Kim, H. Knowledge-grounded chatbot based on dual wasserstein generative adversarial networks with effective attention mechanisms. *Appl. Sci.* **2020**, *10*, 3335.
10. Ghazvininejad, M.; Brockett, C.; Chang, M.W.; Dolan, B.; Gao, J.; Yih, W.t.; Galley, M. A knowledge-grounded neural conversation model. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; Volume 32.
11. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems*; 2017; Long Beach, CA, USA, 4-9 Dec 2017; Volume 30. Available online: https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html (accessed on 25 May 2022).
12. Sukhbaatar, S.; Weston, J.; Fergus, R.; et al. End-to-end memory networks. In *Advances in Neural Information Processing Systems*; Montreal, Canada, 7-12 Dec 2015; Volume 28. Available online: https://proceedings.neurips.cc/paper/2015/hash/8fb21ee7a2207526da55a679f0332de2-Abstract.html (accessed on 25 May 2022).
13. He, H.; Lu, H.; Bao, S.; Wang, F.; Wu, H.; Niu, Z.; Wang, H. Learning to select external knowledge with multi-scale negative sampling. *arXiv* **2021**, arXiv:2102.02096.
14. Feng, S.; Wan, H.; Gunasekara, C.; Patel, S.S.; Joshi, S.; Lastras, L.A. doc2dial: A goal-oriented document-grounded dialogue dataset. *arXiv* **2020**, arXiv:2011.06623.
15. Kim, B.; Ahn, J.; Kim, G. Sequential latent knowledge selection for knowledge-grounded dialogue. *arXiv* **2020**, arXiv:2002.07510.
16. Saha, S.; et al. Open information extraction from conjunctive sentences. In Proceedings of the 27th International Conference on Computational Linguistics, Santa Fe, NM, USA, 20–26 August 2018; pp. 2288–2299.
17. Levi, F. *Finite Geometrical Systems: Six Public Lectures Delivered in February, 1940, at the University of Calcutta*; University of Calcutta: West Bengal, India, 1942.
18. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; Bengio, Y. Graph attention networks. *arXiv* **2017**, arXiv:1710.10903.
19. Srivastava, R.K.; Greff, K.; Schmidhuber, J. Highway networks. *arXiv* **2015**, arXiv:1505.00387.
20. Papineni, K.; Roukos, S.; Ward, T.; Zhu, W.J. Bleu: A method for automatic evaluation of machine translation. In Proceedings of the 40th annual meeting of the Association for Computational Linguistics, Stroudsburg, PA, USA, 7–12 July 2002; pp. 311–318.
21. Chen, B.; Cherry, C. A systematic comparison of smoothing techniques for sentence-level bleu. In Proceedings of the Ninth Workshop on Statistical Machine Translation, Baltimore, MD, USA, 26–27 June 2014; pp. 362–367.
22. Post, M. A call for clarity in reporting BLEU scores. *arXiv* **2018**, arXiv:1804.08771.
23. Kim, B.; Lee, D.; Kim, S.; Lee, Y.; Huang, J.X.; Kwon, O.W.; Kim, H. Document-grounded goal-oriented dialogue systems on pre-trained language model with diverse input representation. In Proceedings of the 1st Workshop on Document-grounded Dialogue and Conversational Question Answering (DialDoc 2021), 5 August 2021; pp. 98–102.
24. Yang, S.; Zhang, R.; Erfani, S. Graphdialog: Integrating graph knowledge into end-to-end task-oriented dialogue systems. *arXiv* **2020**, arXiv:2010.01447.