# G4: Grounding-guided Goal-oriented Dialogues Generation with Multiple Documents

**Shiwei Zhang** [†], **Yiyang Du**[‡], **Guanzhong Liu** [†§], **Zhao Yan**[†], **Yunbo Cao**[†]
[†]Tencent Cloud Xiaowei, [§]Tianjin University
[‡] State Key Lab of Software Development Environment, Beihang University
{zswzhang, zhaoyan, yunbocao}@tencent.com, duyiyang@buaa.edu.cn
rogerliu425@tju.edu.cn

## Abstract

Goal-oriented dialogues generation grounded in multiple documents(MultiDoc2Dial) is a challenging and realistic task. Unlike previous works which treat document-grounded dialogue modeling as a machine reading comprehension task from single document, MultiDoc2Dial task faces challenges of both seeking information from multiple documents and generating conversation response simultaneously. This paper summarizes our entries to agent response generation subtask in MultiDoc2Dial dataset. We propose a three-stage solution, Grounding-guided goal-oriented dialogues generation(G4), which predicts groundings from retrieved passages to guide the generation of the final response. Our experiments show that G4 achieves SacreBLEU score of 31.24 and F1 score of 44.6 which is 60.7% higher than the baseline model.

## 1 Introduction

Conversational question answering techniques have attracted widespread attention as a fusion of document-based question answering and dialogue generation techniques. Most previous works have focused on single-document conversational question answering tasks, such as QuAC(Choi et al., 2018), CoQA(Reddy et al., 2019), Doc2Dial(Feng et al., 2020). As a more realistic task, goal-oriented dialogues generation is based on multiple documents as MultiDoc2Dial(Feng et al., 2021) faces challenges of identifying useful pieces of text from documents and generating response simultaneously.

Inspired by the method of Open-domain question answering (OpenQA), the MultiDoc2Dial task can be solved in a two-stage framework(Robertson et al., 1995; Lewis et al., 2020; Izacard and Grave, 2020a,b): (i) first to retrieve relevant passages from the knowledge source (the retriever)(Jones, 1972; Robertson et al., 1995; Karpukhin et al., 2020; Xiong et al., 2020; Brown et al., 2020); and (ii) second to produce an answer based on retrieved passages and the question (the generator)(Raffel et al., 2019; Lewis et al., 2019). what's more, the end-to-end methods which learn to retrieve and generate simultaneously(Lee et al., 2021; Singh et al., 2021) also achieves good results.

In MultiDoc2Dial task, as the dialogue flow shifts among the grounding documents through the conversation which makes the current session relevant to multiple documents, identifying the content that best matches the question from the relevant documents becomes the biggest challenge. After analysis, it is found that two-stage methods fail to locate the answer span position from multiple relevant documents and generate irrelevant responses to the query.

In this paper, we propose a grounding-guided three-stage framework(Retriever-Reader-Generator). The framework imitates the process of humans looking for answers from a browser: first read each relevant retrieved documents with question, and then combine the understanding of each document to generate a final answer. The generator of third stage can be guided by grounding spans, phrases in the retrieved document predicted by reader, mitigating the excessive deviation of the generated result from the correct response. Since the same corpus data is shared among reader and generator, data distribution of the input for generator are different in training and inference. We also propose a data augmentation approach to alleviate the discrepancy between training and inference. To conclude, our contributions are as follows:

- we propose a grounding-guided three-stage framework that mitigates the deviation of response from the question.
- we present a data augmentation approach which improves the diversity of groundings for generation thus further improving the robustness of the multi-stage framework.

*Proceedings of the Second DialDoc Workshop on Document-grounded Dialogue and Conversational Question Answering*, pages 108 - 114
May 26, 2022 ©2022 Association for Computational Linguistics

## 2 Method

Our proposed model G4 is a three-stage framework: (i) retrieve relevant passages from the knowledge(retriever). (ii) predict groundings based on retrieved passages(reader). (iii)generate a response based on the groundings predicted by reader and relevant passages from retriever(generator).

### 2.1 Problem Definition

Given dialogue history $\{u_1, \ldots, u_{T-1}\}$ and current user's utterance $u_T$, MultiDoc2Dial task produces the response $u_{T+1}$ based on knowledge from a set of relevant documents $D_0 \subseteq D$, where $D$ denotes all knowledge documents. In particular, in the same dialogue session, different utterances might have different related documents. Depending on the form of the answer, two tasks are proposed in MultiDoc2Dial where the first is to identify the grounding document span and the second is to generate agent response. We focus on the second task in this paper.

### 2.2 Retriever

Given dialogue history $\{u_1, \ldots, u_{T-1}\}$, current user's utterance $u_T$ and passages $P = \{p_0, p_1, \ldots, p_M\}$ which are splited from all knowledge documents. The retriever identify the top-k relevant passages $R = \{r_1, r_2, \ldots, r_k\} \subseteq P$, where $P$ is the splited results of documents $D$. In order to distinguish the above $D$ and $P$, we use "document" and "passage" respectively to denote the text of before and after segmentation. In the retrieval stage, we first split the documents into passages and then use dense-based retriever to identify relevant passages.

**Retriever**. We use the ANCE model (Xiong et al., 2020) to retrieve relevant passages from multi-documents. We finetune ANCE model with positive and negtive examples. We choose the golden passage which include the annotated grounding as positive examples. For initial negative examples, we use grounding and last utterance as query respectively to select top 2 retrieval passages except golden passage as hard negative examples, and use current response utterance with dialogue history as query to choose top 5 to 15 from retrieved results as negative examples based on BM25.

### 2.3 Reader

Taking current utterance $u_T$ with dialogue history $\{u_1, \ldots, u_{T-1}\}$ and a retrieved passage $p_i$ as input, the reader predicts grounding span in passage or reject to answer. Sliding window is applied to process long passage.

**Span restrict.** Unlike standard span-based question answering, possible start and end positions of grounding spans are restricted to phrases in the document. The phrase is consecutive sentences labeled in the original document with HTML tag, and the grounding span is one of the phrases in a document.For datasets without grounding labels, grounding span can be constructed according to final response with simple similarity algorithm. To reduce the difficulty of training, we only consider the start and end position of phrase for predict and apply softmax over tokens corresponding these position like previous work(Daheim et al., 2021). Start and end probability are calculated by a linear projection from the last hidden states of encoder:

$$\hat{\boldsymbol{p}}^{\text{ start}} = \sigma(\varphi(H)m_s) \quad \hat{\boldsymbol{p}}^{\text{ end}} = \sigma(\varphi(H)m_s)$$

where $\hat{\boldsymbol{p}}^{\text{ start}}$ and $\hat{\boldsymbol{p}}^{\text{ end}}$ is start and end probability distribution, $H$ is the represation of encoder, $m_s$ denote the mask vector that set to 1 if the start and end position of phrase else 0, $\sigma$ is softmax function and $\varphi(\circ)$ is MLP. The cost function is defined as :

$$J(\boldsymbol{\theta}) = -\frac{1}{T}\sum_{t=1}^{T} \log\left(\hat{\boldsymbol{p}}_{y_t^s}^{\text{start}}\right) + \log\left(\hat{\boldsymbol{p}}_{y_t^e}^{\text{end}}\right)$$

where $T$ is the number of training samples, $y_t^s$ and $y_t^e$ are the true start and end position of the t-th sample.

### 2.4 Grounding-guided generator

To fully leverage the multiple passages identified by the retriever, we adopt Fusion-in-Decoder(FiD) (Izacard and Grave, 2020b) as our response generation model. Based on seq2seq framework, FiD encodes every passage with query independently and decodes all encoded features jointly to generate response. As summed up in FiD, increasing the number of passages from 10 to 100 leads to significantly improvement on different OpenQA datasets. However, in the MultiDoc2Dial dataset, the current session is related to multiple documents, which makes it difficult for generator to identify grounding span for the current utterance. The above problem is exacerbated when the document is split into multi-passages so that the generated response might be irrelevant to grounding and simply increasing the number of passages yields only insignificant improvement.

| Model | F1 | SacreBLEU | METEOR | RougeL | total |
|---|---|---|---|---|---|
| **DPR+RAG** (Lewis et al., 2020) | 34.25 | 19.44 | 33.30 | 31.85 | 118.84 |
| **EMDR** (Singh et al., 2021) | 43.33 | 24.82 | 41.81 | 41.83 | 151.79 |
| **DPR+FiD**(Izacard and Grave, 2020b) | 42.14 | 28.58 | 39.78 | 40.67 | 151.17 |
| **G4-base** *(DPR+G$^{org}$)* | 43.65 | 30.33 | 41.38 | 41.64 | 157.00 |
| **G4-fin** *(ANCE+G$^{org}$)* | 44.11 | 30.91 | 41.85 | 42.11 | 159.39 |
| **G4-aug** *(DPR+G$^{augment}$)* | 44.22 | 30.73 | 41.96 | 42.28 | 159.19 |
| **G4** *(ANCE+G$^{augment}$)* | 44.60 | 31.24 | 42.41 | 42.68 | 160.93 |

Table 1: The results of Different models in MultiDoc2Dial dataset. DPR: DPR model officially released by MultiDoc2Dial. DPR$^{optimized}$: our fine-tuned ANCE model described in 2.2. G$^{org}$/G$^{aug}$: grounding-guided generator described in 2.4. G$^{aug}$: grounding-guided generation with data augmentation while G$^{org}$ without augmentation.

**Encode with grounding span and fusionly decode.** Grounding-guided generator use the grounding span predicted by reader of second stage to guide the generator to produce agent response. Based on Fusion-in-Decoder model, we proposed two approach to introduce grounding span to guide generator:(i)The first way is to directly concatenate the grounding span to the front of the original passage. The input form of FiD encoder can be described as follows:

$$[\bar{S}_u, u_T, u_{T-1}...u_1; \bar{S}_g, g; \bar{S}_p, p]$$

where $u_t$, $g$, $p$ is utterance of turn $t$, grounding span, original passage respectively and all parts start with special token: $\bar{S}_u$, $\bar{S}_g$,$\bar{S}_p$. In particular, $g$ will be set to the empty string while reader reject to answer for current passage. (ii)The second way is to tag start and end position of grounding span in the passage. The input form listed as follows:

$$[\bar{S}_u, u_T, u_{T-1}...u_1; \bar{S}_p, p_i^0, p_i^1...\bar{S}_g, p_i^s, ...p_i^e, \bar{E}_g, p_i^n]$$

where $p_i = \{p_i^0, \dots, p_i^n\}$ is the original retrieved passages, $\{p_i^s, \dots, p_i^e\}$ is grounding span predicted by reader in the passage with the start and end position $\bar{S}_g$ $\bar{E}_g$ and $\bar{S}_g$ $\bar{E}_g$ will be deleted if reader reject to answer. In particular, when the length of the passage exceeds the maximum limit of the encoder, we also use dynamic truncate to ensure that the grounding spans are within the encoding window and are not truncated. If the length is greater than the maximum encoding window length, above (i) truncate from the tail, while (ii) truncate from head and tail to ensure that the grounding span is located in the middle of the window as much as possible. In the first method, if the length is greater than the maximum encoding window length in a conventional way, in the second method, the grounding span is truncated to ensure that the grounding span is located in the center of the window as much as possible.

**Data augmentation.** Since the generator and reader share the same corpus, the reader needs to predict on its training set. The F1 score of grounding prediction is 0.98 on the training set and 0.79 on the evaluation set, which would cause the generator to be overconfident in the given grounding and underperform on the evaluation set. To alleviate the above problem and enhance the robustness of the model, the reader accepts the top-k passages from retriever and finds evidence as "grounding span" for every passage include negative retrieved passage, then the generator produce the final response with evidence from reader. What's more, we adopt two methods of data augmentation for generator in training phase: (i) The first method replaces the groundings spans predicted randomly by the reader with another span of the same length with probability $p$. (ii) The second way replace the groundings spans predicted with one of the $n$ best predicted spans predicted by reader with a probability $p$.

## 3 Experiments

**Data** We use the MultiDoc2Dial dataset(Feng et al., 2021) , a new task and dataset on modeling goal-oriented dialogues grounded in multiple documents, which contains 29748 queries in 4796 dialogues grounded in 488 documents.

**Baseline** Considering that MultiDoc2Dial is a relatively new benchmark, we tried several retriever-reader architecture models. We compare our model with the baseline model RAG(Lewis et al., 2020) in the MultiDoc2Dial paper. FiD(Izacard and Grave, 2020b) is a two-stage pipeline method which first retrieves passages and performs evidence fusion in the decoder based on multiple passage. EMDR(Singh et al., 2021) is the state-of-the-

| Model Variant | F1 | SacreBLEU | METEOR | RougeL | total |
|---|---|---|---|---|---|
| **G4-base** *(DPR+$G^{org}$)* | 43.65 | 30.33 | 41.38 | 41.64 | 157.00 |
| w/o grounding | 42.14 | 28.58 | 39.78 | 40.67 | 151.17 |
| w grounding$^{\text{concat}}$ | 43.13 | 29.48 | 40.79 | 41.23 | 154.63 |
| w grounding$^{\text{tag}}$ | 43.65 | 30.33 | 41.38 | 41.64 | 157.00 |
| w augment$^{\text{nbest}}$ | 44.09 | 30.63 | 41.44 | 42.09 | 158.25 |
| w augment$^{\text{random}}$ | 44.22 | 30.73 | 41.96 | 42.28 | 159.19 |

Table 2: The ablation results of G4 on MultiDoc2Dial validation dataset. w/*o* grounding: don't use the reader module, the generator directly accept the concatenation of the retrieved passages. grounding$^{\text{concat}}$: add grounding span with concatenate method. grounding$^{\text{tag}}$: add grounding span with tag method. augment$^{\text{nbest}}$: noise data with n-best predicted spans($n = 20$). augment$^{\text{random}}$: random noise data.

| Model | R@1 | R@5 | R@10 |
|---|---|---|---|
| **BM25** | 17.26 | 37.80 | 46.49 |
| **DPR$^{\text{official}}$** | 38.40 | 65.90 | 75.20 |
| **DPR$^{\text{aug}}$** | 42.78 | 67.98 | 77.05 |
| **ANCE** | 39.54 | 68.46 | 77.27 |

Table 3: Performance of different retrieve methods on MultiDoc2Dial validation dataset. DPR$^{\text{official}}$: official DPR finetuned by (Feng et al., 2021). ANCE: ANCE model described in 2.4. DPR$^{\text{aug}}$: our DPR trained with better negative examples as ANCE.

art model for OpenQA task on the Natural Question dataset(Kwiatkowski et al., 2019) which apply an end-to-end training method for documents retrieval and answer generation.

**Implementation** In retriever, we choose the positive passage with 2 negative and 2 hard-negative examples to train ANCE model. We retrieve 50 passages for reader and generator and set batch size to 128. In reader, we initialize our span-based machine reading comprehension with RoBERT-based model and batch size is 32. In generator stage, we adopt the Fusion-in-Decoder model and follow it's architectural and basic experimental settings. We choose the T5-base as the initial weights and set the max source(dialogue+passage) length to 512 while max answer(response) length to 50. We use the probability of $p = 0.3$ to add noisy data mentioned in 2.4. Other experiment hyper-parameters can be seen in Appendix A.

**Results.** Table 1 reports the evaluation results on MultiDoc2Dial validation. We observe that our grounding-guided method (G4-base) clearly outperforms the MultiDoc2Dial baseline. Compared with the two-stage model FiD, the SacreBLEU score is significantly improved by 1.75 points, which fully proves that the grounding span predicted by the reader from second stage can effectively improve the generator's performance. Our grounding-guided generator with data augmentation and better retriever can even further improve by 2.66 SacreBLEU score than baseline. Combining Table 1 and Table 3, it can be concluded that improving the recall of retriever in the first stage can effectively improve the final generation. By choosing better negative examples, both our trained DPR and ANCE models achieve better results. At the same time, we also noticed that the end-to-end training method EMDR performs not very well on MultiDoc2Dial task.

Table 2 shows the effect of different methods of grounding-guided generation and data augmentation. Our two methods of introducing grounding span have significantly improved generation result, the first concatenation-based method by 0.99 points SacreBLEU and second tag-based by 1.51 points SacreBLEU. The tag-based method not only tells encoder the position of grounding span, but also dynamically adjusts the window according to the position of the grounding span to ensure that important part of passage wouldn't be truncated when the passage exceeds the maximum length of the encoder. Therefore, the tag-based method can achieve better results than concatenation-based method. In the training phase, adding noise data to grounding predicted by reader can improve the robustness of generator and avoid the generator over-reliance on the prediction results of the reader. Using the random selection for noise outperforms n-best data predicted by the model, probably because n-best answers are very similar to grounding spans.

What's more, since the retrieval stage may mistakenly recall irrelevant documents to the query, we also experimented with making the reader to identify negative retrieved passages. By randomly selecting retrieved passages that do not contain grounding span as negative samples, we train a binary classifier to enables the reader to identify irrelevant passages. In the inference phase, model will reject to predict grounding span when the binary classifier identifies a retrieved passage as negative, otherwise adopt the grounding span as final answer. The negative passages predicted by reader are also used as the input of generator, but without grounding span. The result shows that the reader model with the ability to identify negative passages has no gain or even worse for the final response generation. We believe that the reason may be that the reader and generator share the same training corpus, and training data with results predicted by reader makes the generator prone to overfitting.

## 4 Conclusion

In this paper, we propose a three-stage approach to the MultiDoc2Dial task, which adds readers to a two-stage framework based on retriever and generator. We show that the grounding predicted by the reader can effectively mitigate the deviation of the generated result from the grounding and correct response. In future work, we plan to introduce grounding information in a more efficient way based on end-to-end models.

## References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. Quac: Question answering in context. *arXiv preprint arXiv:1808.07036*.

Nico Daheim, David Thulke, Christian Dugast, and Hermann Ney. 2021. Cascaded span extraction and response generation for document-grounded dialog. *arXiv preprint arXiv:2106.07275*.

Song Feng, Siva Sankalp Patel, Hui Wan, and Sachindra Joshi. 2021. Multidoc2dial: Modeling dialogues grounded in multiple documents. In *EMNLP*.

Song Feng, Hui Wan, Chulaka Gunasekara, Siva Sankalp Patel, Sachindra Joshi, and Luis A Lastras. 2020. doc2dial: A goal-oriented document-grounded dialogue dataset. *arXiv preprint arXiv:2011.06623*.

Gautier Izacard and Edouard Grave. 2020a. Distilling knowledge from reader to retriever for question answering. *arXiv preprint arXiv:2012.04584*.

Gautier Izacard and Edouard Grave. 2020b. Leveraging passage retrieval with generative models for open domain question answering. *arXiv preprint arXiv:2007.01282*.

Karen Sparck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Haejun Lee, Akhil Kedia, Jongwon Lee, Ashwin Paranjape, Christopher D Manning, and Kyoung-Gu Woo. 2021. You only need one model for open-domain question answering. *arXiv preprint arXiv:2112.07381*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Siva Reddy, Danqi Chen, and Christopher D Manning. 2019. Coqa: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. 1995. Okapi at trec-3. *Nist Special Publication Sp*, 109:109.

Devendra Singh, Siva Reddy, Will Hamilton, Chris Dyer, and Dani Yogatama. 2021. End-to-end training of multi-document reader and retriever for open-domain question answering. *Advances in Neural Information Processing Systems*, 34.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *arXiv preprint arXiv:2007.00808*.

# A Experiment Hyper-parameters

## A.1 Hyper-parameters for retriever

```
train_batch_size=128
num_negtive_examples=2
num_hard_negtive_examples=2
top_k=50
max_query_length=512
max_passage_length=512
dropout=0.1
attention_dropout=0.1
optim=adam
learning_rate=2e-05
```

## A.2 Hyper-parameters for reader

```
train_batch_size=32
eval_batch_size=4
doc_stride=128
max_seq_length=512
max_ans_length=128
initial_weight=roberta-base
optim=adam
warmup_steps=1000
learning_rate=3e-5
```

## A.3 Hyper-parameters for generator

```
train_batch_size=4
n_passages=50
max_source_length=512
max_target_length=50
dropout=0.1
attention_dropout=0.1
initial_weight=T5-base
learn_rate=1e-04
gradient_accumulation_steps=2
```