Grounded Dialogue Generation with Cross-encoding Re-ranker, Grounding Span Prediction, and Passage Dropout

Kun Li[1]*, Tianhua Zhang[2]*, Liping Tang[2], Junan Li[2], Hongyuan Lu[1], Xixin Wu[1], Helen Meng[1,2]†
[1]The Chinese University of Hong Kong, Hong Kong SAR, China
[2]Centre for Perceptual and Interactive Intelligence, Hong Kong SAR, China
{kunli, hylu, hmmeng}@se.cuhk.edu.hk
{thzhang, lptang, jali}@cpii.hk
xixinwu@cuhk.edu.hk

Abstract

MultiDoc2Dial presents an important challenge on modeling dialogues grounded with multiple documents. This paper proposes a pipeline system of "retrieve, re-rank, and generate", where each component is individually optimized. This enables the passage re-ranker and response generator to fully exploit training with ground-truth data. Furthermore, we use a deep cross-encoder trained with localized hard negative passages from the retriever. For the response generator, we use grounding span prediction as an auxiliary task to be jointly trained with the main task of response generation. We also adopt a passage dropout and regularization technique to improve response generation performance. Experimental results indicate that the system clearly surpasses the competitive baseline and our team CPII-NLP ranked 1st among the public submissions on ALL four leaderboards based on the sum of F1, SacreBLEU, METEOR and RougeL scores.

1 Introduction

The task of developing information-seeking dialogue systems has seen many recent research advancements. The goal is to answer users' questions grounded on documents in a conversational manner. MultiDoc2Dial[1] is a realistic task proposed by Feng et al. (2021) to model goal-oriented information-seeking dialogues that are grounded on multiple documents and participants are required to generate appropriate responses towards users' utterances according to the documents. To facilitate this task, the authors also propose a new dataset that contains dialogues grounded in multiple documents from four domains. Unlike previous work that mostly describe document-grounded dialogue modeling as a machine reading comprehension task based on one particular document or passage, the MultiDoc2Dial involves multiple topics within a conversation, hence it is grounded on different documents. The task contains two sub-tasks: Grounding Span Prediction aims to find the most relevant span from multiple documents for the next agent response, and Agent Response Generation generates the next agent response. This paper focuses on our work in to the second sub-task, and presents three major findings and contributions:

- In order to fully leverage the ground-truth training data, we propose to individually optimize the retriever, re-ranker, and response generator.
- We propose to adopt a deep cross-encoded re-ranker that is trained with localized hard negatives sampled from the retriever results.
- We propose to use grounding span prediction as an auxiliary task for the generator and use passage dropout as a regularization technique to improve the generation performance.

Experimental results indicate that our proposed system achieves a performance with marked improvement over the strong baseline.

2 Related Work

Open-domain Question Answering systems have evolved to adopt the popular "Retriever-Reader (Generator)" architecture since DrQA (Chen et al., 2017). Previous work (Lee et al., 2019, Guu et al., 2020) adopt end-to-end training strategy to jointly learn the retriever and reader with question-answer pairs. Retrieval-augmented Generation (RAG) (Lewis et al., 2020b) uses Dense Passage Retriever (DPR) (Karpukhin et al., 2020) as the retriever to extract multiple documents related to the query and feed them into a BART (Lewis et al., 2020a) generator for answer generation. Izacard and Grave (2021) proposed the Fusion-in-Decoder method

*Contributed equally.
†Corresponding author.
[1]https://doc2dial.github.io/multidoc2dial/

Proceedings of the Second DialDoc Workshop on Document-grounded Dialogue and Conversational Question Answering, pages 123 - 129

which processes passages individually in the encoder but jointly in the decoder, surpassing the performance of RAG.

Other work like QuAC (Choi et al., 2018), ShARC (Saeidi et al., 2018) and CoQA (Reddy et al., 2019) focus on the machine reading comprehension task, which assumes that the associated document is given. In particular, Feng et al. (2020) proposed the Doc2Dial task ,which aims to extract the related span from the given documents for generating the corresponding answer.

3 Task Description

The MultiDoc2Dial task aims to generate an appropriate response $\mathcal{R}$ based on an input query $\mathcal{Q}$ (the current user turn u_T and the concatenated dialogue history $\{u_1^{T-1}\} := u_1, u_2, ..., u_{T-1}$) and a collection of passages $\{\mathcal{P}_i\}_{i=1}^{M}$. The passages are extracted from documents based on document structural information indicated by markup tags in the original HTML file. The organizer splits the MultiDoc2Dial data into train, validation, development and test set, and results on the latter two are evaluated through the leaderboard[2]. The validation, development and test set contain two settings: *seen* and *unseen*, which is categorized based on whether there are dialogues grounded on the documents seen/unseen during training. We leave detailed dataset description in Appendix A.

4 Methodology

We propose a pipeline system of "retrieve, re-rank, and generate". Following previous work in Lewis et al. (2020b); Feng et al. (2021), we adopt DPR (Karpukhin et al., 2020) as the retriever (§4.1) to efficiently filter out irrelevant passages and narrow the search space. We then refine the retrieval results with a deep cross-encoder (§4.2) trained with localized negatives (Gao et al., 2021). We introduce a passage dropout and regularization technique to enhance the robustness of the generator (§4.3) and use the grounding span prediction as an auxiliary task. Further more, pipeline training is adopted where each component is individually optimized to fully utilize the supervision. Experimental results (§5.3) also indicate the effectiveness and merits of the training strategy, which we observed to be a key factor for the performance gain.

[2]https://eval.ai/web/challenges/challenge-page/1437/leaderboard

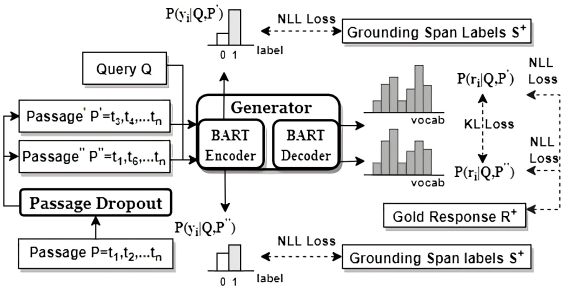

Figure 1: Training process of our generator.

4.1 Passage Retrieval

Following Feng et al. (2021), we adopt DPR (Karpukhin et al., 2020) as the retriever with a representation-based bi-encoder, that is, a dialogue query encoder $q(\cdot)$ and a passage context encoder $p(\cdot)$. Given an input query $\mathcal{Q}$ and a collection of passages $\{P_i\}_{i=1}^{M}$, we extract the query encoding as $q(\mathcal{Q})$ and the passage encoding as $p(\mathcal{P}_i)$. The similarity is defined as the dot product of the two vectors $\langle q(\mathcal{Q}), p(\mathcal{P}_i)\rangle$ and the model is trained to optimize the negative log likelihood of the positive passage among L in-batch and hard negatives. We then pre-compute the representations of all passages and index them offline. Maximum Inner Product Search (MIPS) with Faiss (Johnson et al., 2017) is adopted to retrieve the top-K passages during inference.

4.2 Passage Re-ranking

To re-rank the passages retrieved by DPR, we use a BERT-based cross-encoder that exploits localized negatives sampled from DPR results (Gao et al., 2021). This means that the construction of the training set for the re-ranker is based on the top negative passages retrieved by the DPR. Specifically, given a query $\mathcal{Q}$, its corresponding ground truth passage $\mathcal{P}^+$, and its top-N negative passages $\{\mathcal{P}_j^-\}_{j=1}^{N}$ retrieved by DPR, we first calculate a deep distance function for each positive and negative passage against the query:

$$\text{dist}(\mathcal{Q}, \mathcal{P}) = v^T \text{cls}(\text{BERT}(\text{concat}(\mathcal{Q}, \mathcal{P}))), \quad (1)$$

where v represents a trainable vector, cls extracts the [CLS] vector from BERT. Consequently, such a distance function is deeply cross-encoded, as we feed the concatenation of the query and the passage into the model instead of encoding them individually with a representation-based bi-encoder (Feng

et al., 2021). We then apply a contrastive loss:

$$\mathcal{L}_c = -\log \frac{\exp(\text{dist}(\mathcal{Q}, \mathcal{P}^+))}{\sum_{\mathcal{P} \in \mathcal{P}_\pm} \exp(\text{dist}(\mathcal{Q}, \mathcal{P}))}, \quad (2)$$

where $\mathcal{P}_\pm$ represents $\mathcal{P}^+ \cup \{\mathcal{P}_i^-\}_{i=1}^N$. Here, it is important to condition the gradient on the negative passages to learn to recognize the positive passage from hard negatives retrieved by the DPR. [3]

Ensemble We create an ensemble of three pre-trained models (Dietterich, 2000), namely, BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), and ELECTRA (Clark et al., 2020) for re-ranking. We first calculate their distance function with Equation 1, with the output scores denoted as $\mathcal{O}_B$, $\mathcal{O}_R$, and $\mathcal{O}_E$. We define the final scores $\mathcal{O}$ as the weighted summation of the above three scores:

$$\mathcal{O} = \alpha \mathcal{O}_B + \beta \mathcal{O}_R + \gamma \mathcal{O}_E, \quad (3)$$

where α, β, and γ represent the weight hyper-parameters for each model.

4.3 Response Generation

For response generation, we leverage the pre-trained sequence-to-sequence model $\text{BART}_{\text{large}}$ (Lewis et al., 2020a), where the encoder is fed the concatenation of a query and a passage $[\mathcal{Q}, \mathcal{P}]$, and the decoder is then required to generate the corresponding response $\mathcal{R}$. We use the ground truth passage as $\mathcal{P}$ for training. The training process can be summarized as follows:

Joint Training with Grounding Prediction The grounding span in a passage is the supporting evidence for the response, which can provide helpful information for response generation. Therefore, we take grounding span prediction as the auxiliary task and apply multi-task learning for model training. Specifically, the passage is first encoded into a sequence of hidden representations $h_i = \text{Encoder}([\mathcal{Q}, \mathcal{P}]), i \in \{1, ..., |\mathcal{P}|\}$. Then a classifier outputs the probability of the i-th token of $\mathcal{P}$ to lie within the grounding span as $P(y_i|\mathcal{Q}, \mathcal{P}) = \text{sigmoid}(\text{MLP}(h_i))$. We define this task's training objective as:

$$\mathcal{L}_G = -\sum_{i=1}^{|\mathcal{P}|} \log P(y_i|\mathcal{Q}, \mathcal{P}). \quad (4)$$

[3] Feng et al. (2021) found that there exists passages that are similar to one another in the dataset. Therefore, it is intuitively important to distinguish these hard negative passages from the ground truth passage. Empirically, we also found that excluding hard negative passages from the training process hampers the re-ranking performance.

Passage Dropout and Regularization Preliminary experiments indicate that the generator is prone to overfit to some passages quoted frequently in the train set, which may cause generalization errors when applied to previously unseen passages. Hence, we apply passage dropout to enhance the robustness of the generator. In details, for a training sample $([\mathcal{Q}, \mathcal{P}], \mathcal{R})$, a consecutive span with a specified length (of 25% in our experiments) in $\mathcal{P}$ is randomly selected and then dropped, which produces $\mathcal{P}'$. It is noteworthy that passage dropout is required to avoid truncating content of grounding spans.[4] Furthermore, we repeat passage dropout twice for each sample in a batch, and obtain $([\mathcal{Q}, \mathcal{P}'], \mathcal{R})$ as well as $([\mathcal{Q}, \mathcal{P}''], \mathcal{R})$. Since the grounding span in a passage serves as the oracle for response generation, the two modified inputs should have similar prediction distribution, denoted as $P(r_i|\mathcal{Q}, \mathcal{P}', r_{<i})$ and $P(r_i|\mathcal{Q}, \mathcal{P}'', r_{<i})$, where r_i is the i-th token of $\mathcal{R}$. Hence, inspired by Liang et al. (2021), we propose to regularize the predictions from different passage dropouts by minimizing the bidirectional Kullback-Leibler (KL) divergence between these two different output distributions as $\mathcal{L}_{KL}$:

$$\sum_i (\text{KL}(P(r_i|\mathcal{Q}, \mathcal{P}', r_{<i}) \| P(r_i|\mathcal{Q}, \mathcal{P}'', r_{<i})) + \text{KL}(P(r_i|\mathcal{Q}, \mathcal{P}'', r_{<i}) \| P(r_i|\mathcal{Q}, \mathcal{P}', r_{<i}))). \quad (5)$$

We define the training objective for response $\mathcal{R}$ as the basic negative log-likelihood:

$$\mathcal{L}_{NLL} = -\sum_i (\log P(r_i|\mathcal{Q}, \mathcal{P}', r_{<i}) + \log P(r_i|\mathcal{Q}, \mathcal{P}'', r_{<i})). \quad (6)$$

With passage dropout, the learning objective of grounding prediction (Eq.4) is updated for $\mathcal{P}'$ and $\mathcal{P}''$. Then we have the final training objective:

$$\mathcal{L} = \frac{1}{2}\mathcal{L}_{KL} + \mathcal{L}_{NLL} + \mathcal{L}_G. \quad (7)$$

4.4 Inference

After the re-ranker returns the top-5 passages corresponding to the query $\mathcal{Q}$, we filter out the passages with a low re-ranking score (Eq.3), namely, the ones that have a score gap of over 0.3 comparing to the top-1. Then the remaining passages are concatenated as a single passage $\mathcal{P}$. Finally the generator

[4] If the selected span overlaps with a grounding span, this sampling is discarded and another span would be sampled.

seen	Val			Dev			Test		
	F1	**S-BLEU**	**ROUGE**	**F1**	**S-BLEU**	**ROUGE**	**F1**	**S-BLEU**	**ROUGE**
RAG	36.64	23.24	35.23	36.23*	21.41*	34.01*	35.85*	22.26*	33.82*
Ours	47.29	34.29	46.04	50.14	34.99	47.91	52.06	37.41	50.19
unseen	Val			Dev			Test		
	F1	**S-BLEU**	**ROUGE**	**F1**	**S-BLEU**	**ROUGE**	**F1**	**S-BLEU**	**ROUGE**
RAG	13.68	4.46	13.19	18.66*	5.99*	16.95*	19.26*	6.32*	17.16*
Ours	36.74	24.20	35.49	36.39	26.33	34.71	34.65	27.57	34.49

Table 1: Comparison between the baseline and the proposed framework on the validation, development and test set. The scores with * are cited from the leaderboard. **S-BLEU** represents SacreBLEU.

predicts a response $\mathcal{R}$ given the input $[\mathcal{Q}, \mathcal{P}]$.[5] We employ beam-search (beam width=5) during decoding.

5 Experiments and Results

We evaluate the passage retrieval results with recall (R) and mean reciprocal rank (MRR). We report response generation performance based on F1, Exact Match (EM) (Rajpurkar et al., 2016), SacreBLEU (S-BLEU; Post, 2018), and RougeL (Lin, 2004).

5.1 Main Results

Table 1 shows the results we obtain for each data split, each including the *seen* and *unseen* settings. RAG (Lewis et al., 2020b) is the baseline adopted by the organizer, and we reproduce it with a more aggressive setting (e.g., a greater input length and beam size), in order to have a fair comparison with the proposed approach. Our generator is a single model. Table 1 shows that the proposed approach consistently outperforms the baseline with significant gaps. We argue that the improvement is derived from (1) high-quality retrieval, (2) stronger generator and (3) pipeline-based training, which will be discussed in the following sections.

5.2 Retrieval Results

Since the passage supervision of the development and test data is unavailable and the leaderboards do not provide the retrieval scores, we analyze the passage retrieval performance on the validation set[6] as shown in Table 2. The baseline adopts DPR (Karpukhin et al., 2020) as retriever, and we evaluate both the official and our reproduced versions.

[5]Grounding Prediction and passage dropout are not implemented in the inference phrase.

[6]We evaluate on a cleaned validation set where repeated queries are removed, resulting in 4181 unique instances (cf. 4201 originally) and 121 unique instances (cf. 121 originally) in the *seen* and *unseen* settings respectively.

Method	*seen*			*unseen*		
	MRR@5	**R@1**	**R@5**	**MRR@5**	**R@1**	**R@5**
Official DPR*	0.487	0.379	0.656	0.277	0.207	0.405
Reproduced DPR	0.548	0.445	0.714	0.328	0.248	0.471
BERT B	0.719	0.643	0.834	0.615	0.529	0.752
ELECTRA E	0.719	0.640	0.837	0.582	0.521	0.694
RoBERTa R	0.748	0.683	0.849	0.641	0.562	0.760
$\mathcal{E}(B, R)$	0.754	0.689	0.855	0.664	0.603	**0.769**
$\mathcal{E}(E, R)$	0.756	0.689	**0.858**	0.643	0.595	0.719
$\mathcal{E}(B, E, R)$	**0.760**	**0.696**	**0.858**	**0.666**	**0.620**	0.744

Table 2: Retrieval performance on the MultiDoc2Dial validation set. All models are fine-tuned using the training set only. * indicates the model trained on the official pre-processed data; others are trained on our pre-processed version. $\mathcal{E}(\cdot)$ denotes ensemble.

Introducing the re-ranker gave marked improvement for all three pre-trained models, especially when applied to the *unseen* passages. In particular, RoBERTa achieves 53.5% and 126.6% improvement over the Reproduced DPR at R@1 on the *seen* and *unseen* settings respectively. The ensemble of different re-rankers brings further improvement – $\mathcal{E}(B, E, R)$ exceeds the best single re-ranker by around 0.01 across all metrics on the *seen* data. Furthermore, improved retrieval directly enhances the final task results. Besides a more powerful generator, the large gap between RAG and our approach on the *unseen* Val data in Table 1 may also be attributed to the great performance gain on passage retrieval, from 0.248 to 0.62 on R@1.

5.3 Ablation Study on the Generator

Table 3 shows that each component in our approach contributes to improvement. Passage dropout and regularization bring notable performance gains for the *unseen* setting. This demonstrates robustness in the generator, which is important in practical use.

To investigate the merits of pipeline training on generation, we separate the $\text{BART}_{\text{large}}$ generator from other parts in the reproduced RAG. We input queries combined with the passages returned by the re-reranker for inference. The first and sec-

Method	*seen*			*unseen*		
	F1	**EM**	**S-BLEU**	**F1**	**EM**	**S-BLEU**
BART in the RAG	43.77	6.36	30.91	31.92	2.48	21.25
BART	45.91	7.02	32.36	32.93	2.48	20.73
+ multi-task training	46.51	6.67	32.90	33.61	2.48	21.37
+ passage dropout	47.05	**7.38**	32.82	34.27	4.13	21.94
+ regularization	**47.29**	7.31	**34.29**	**36.74**	**4.96**	**24.20**

Table 3: Ablation analysis of the generators based on the validation set. *BART in the RAG* denotes the generator in the fully-trained RAG. The same retrieval is used in all cases. **S-BLEU** represents SacreBLEU.

ond rows of Table 3 show that the BART in the RAG gained some improvement through better retrieval, but remains inferior to the BART trained in a pipeline fashion. This is mainly attributed by the fact that under the end-to-end training framework of the RAG, the generator could receive some deteriorated query-passage pairs during training, if the retriever can not successfully return gold passages to the generator. Contrarily, pipeline training for the generator can make full use of training data.

6 Conclusion

This paper presents a pipeline system of "retrieve, re-rank, and generate" for the MultiDoc2Dial challenge. The advantage is that each of the three components can fully exploit the ground-truth training data. We apply a deep cross-encoder architecture where we create a training set using localized hard negatives sampled from the retriever results. We adopt grounding span prediction as an auxiliary task to be jointly trained with the response generator. We also apply passage dropout and regularization to improve response generation performance. Experimental results indicate that the proposed system improves over a strong, competitive baseline and our team got 1st place on ALL four leaderboards.

Acknowledgements

This work is partially supported by the Centre for Perceptual and Interactive Intelligence (CPII) Ltd., a CUHK-led under the InnoHK scheme of Innovation and Technology Commission; and in part by the HKSAR RGC GRF (Ref No. 14207619).

References

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada. Association for Computational Linguistics.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. Quac: Question answering in context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2174–2184. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Pre-training transformers as energy-based cloze models. In *EMNLP*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Thomas G. Dietterich. 2000. Ensemble methods in machine learning. In *Multiple Classifier Systems*, pages 1–15, Berlin, Heidelberg. Springer Berlin Heidelberg.

Song Feng, Siva Sankalp Patel, Hui Wan, and Sachindra Joshi. 2021. MultiDoc2Dial: Modeling dialogues grounded in multiple documents. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6162–6176, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Song Feng, Hui Wan, R. Chulaka Gunasekara, Siva Sankalp Patel, Sachindra Joshi, and Luis A. Lastras. 2020. doc2dial: A goal-oriented document-grounded dialogue dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 8118–8128. Association for Computational Linguistics.

Luyu Gao, Zhuyun Dai, and Jamie Callan. 2021. Rethink training of BERT rerankers in multi-stage retrieval pipeline. In *Advances in Information Retrieval - 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28 - April 1, 2021, Proceedings, Part II*, volume 12657 of *Lecture Notes in Computer Science*, pages 280–286. Springer.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. REALM: retrieval-augmented language model pre-training. *CoRR*, abs/2002.08909.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th*

Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021, pages 874–880. Association for Computational Linguistics.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2017. Billion-scale similarity search with GPUs. *arXiv e-prints*, page arXiv:1702.08734.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. *CoRR*, abs/1906.00300.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020a. 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, ACL 2020, Online, July 5-10, 2020*, pages 7871–7880. Association for Computational Linguistics.

Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020b. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Xiaobo Liang, Lijun Wu, Juntao Li, Yue Wang, Qi Meng, Tao Qin, Wei Chen, M. Zhang, and Tie-Yan Liu. 2021. R-drop: Regularized dropout for neural networks. In *Advances in Neural Information Processing Systems*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv e-prints*, page arXiv:1907.11692.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. Coqa: A conversational question answering challenge. *Trans. Assoc. Comput. Linguistics*, 7:249–266.

Marzieh Saeidi, Max Bartolo, Patrick S. H. Lewis, Sameer Singh, Tim Rocktäschel, Mike Sheldon, Guillaume Bouchard, and Sebastian Riedel. 2018. Interpretation of natural language rules in conversational machine reading. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2087–2097. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.

Split	Setting	Instance Num	Passage Num
Train	seen	21451	3820
Validation	*seen*	4201	3820
	unseen	121	963
Development	*seen*	199	3820
	unseen	417	963
Test	*seen*	661	3820
	unseen	126	963

Table 4: Data statistics of different splits. We split a single conversation into multiple instances of the train and validation set.

A Dataset Description

MultiDoc2Dial contains 4796 conversations with an average of 14 turns grounded in 488 documents from four domains. After splitting, the number of passages in the *seen* set is $M = 4110$ for the official data pre-processing and $M = 3820$ for our processed data to remove duplicate passages. Similarly, the number of passages in the *unseen* set is $M = 963$. Table 4 shows the statistics of dataset in different splits.

B Implementation Details

Our implementations of DPR, BERT, RoBERTa, ELECTRA, and BART are based on the Transformers library (Wolf et al., 2019). All the models are trained on an RTX 3090 GPU with 24GB VRAM.

Retriever We train the retriever on our pre-processed MultiDoc2Dial data with an effective batch size of 16 following Facebook DPR (Karpukhin et al., 2020) and the corresponding results are shown in Table 2 named as Reproduced DPR. The Official DPR in Table 2 is fine-tuned with a batch size 128 by the organizer.

Re-ranker Three public pre-trained language models are ensembled, namely, deepset/bert-large-uncased-whole-word-masking-squad2[7], deepset/roberta-large-squad2[8] and deepset/electra-base-squad2[9]. We train the models with a batch size 1 for LARGE (gradient accumulation=4) and 4 for BASE. We use 6 epochs, a learning rate of 1e-5 and weight decay of 0.01. The maximum length of query, i.e., the concatenated dialogue history $\{u_1^{T-1}\}$ and the current user turn u_T is set as 128. Following Feng et al. (2021), the query is constructed using reverse conversation order as $u^T[SEP]agent : u^{T-1}||user : u^{T-2}||...||user : u^1$ and truncated from the tail by the tokenizers. The number of localized negatives in training is 7, sampled from Top-N (N=50) returned negative passages from retriever. During inference, re-ranker re-scores Top-K (K=100) returned passage candidates from retriever and selects the Top-5 passages for generator.

[7]https://huggingface.co/deepset/bert-large-uncased-whole-word-masking-squad2

[8]https://huggingface.co/deepset/roberta-large-squad2

[9]https://huggingface.co/deepset/electra-base-squad2