Long-form Question Answering: An Iterative Planning-Retrieval-Generation Approach

Anonymous ACL submission

Abstract

Long-form question answering (LFQA) poses a challenge as it involves generating detailed answers in the form of paragraphs, which go beyond simple yes/no responses or short factual answers. While existing QA models excel in questions with concise answers, LFQA requires handling multiple topics and their intricate relationships, demanding comprehensive explanations. Previous attempts at LFQA focused on generating long-form answers by utilizing relevant contexts from a corpus, relying solely on the question itself. However, they overlooked the possibility that the question alone might not provide sufficient information to identify the relevant contexts. Additionally, generating detailed long-form answers often entails aggregating knowledge from diverse sources. To address these limitations, we propose an LFQA model with iterative Planning, Retrieval, and Generation. This iterative process continues until a complete answer is generated for the given question. From an extensive experiment on both an open domain and a technical domain QA dataset, we find that our model outperforms the state-of-the-art models on various textual and factual metrics for the LFQA task.

1 Introduction

Question answering (QA) is a computational task that involves providing a relevant and accurate response to a question expressed in natural language. A considerable amount of progress has been made in open-domain question answering, specifically in settings where questions are answerable with short phrases and entities. For example, significant advancements have been made in factoid question-answering (QA) research, which has yielded impressive results with the creation of comprehensive datasets like SQuAD (Rajpurkar et al., 2018) and MS MARCO (Nguyen et al., 2016), as well as the utilization of transformer-based models such as ALBERT (Lan et al., 2019). In many cases, these models have even demonstrated the ability to outperform human performance. However, while short-form question answering has proven effective for simple factual questions, it often falls short when it comes to complex and nuanced questions requiring more comprehensive and detailed responses.

One of the main challenges of LFQA is that there is not much data available for learning this task. One prominent dataset used for this purpose is the ELI5 dataset (Fan et al., 2019). It comprises questions asked on the "Explain Like I'm Five" Reddit forum, along with corresponding answers in paragraph form. However, the questions in ELI5 tend to be broad (e.g., "How do animals see different colors?"), and multiple valid approaches can be taken to answer them. This multiplicity makes it difficult to establish objective criteria for assessing the quality of answers. In a study by (Krishna et al., 2021), various hurdles in effectively leveraging this dataset for meaningful advancements in modeling were highlighted, including the absence of reliable evaluation metrics.

In addition to the limitation of available datasets, the existing models for LFQA exhibit performance shortcomings. For example, the KILT benchmark, recently introduced by (Petroni et al., 2020), is a framework that evaluates retrieval-augmented models on various knowledge-intensive tasks, such as ELI5. It assesses LFQA models based on the quality of their generated answers (measured using ROUGE-L against reference answers) as well as the relevance of retrieved documents (measured using R-precision against human-annotated relevant documents). However, the utilization of retrieved contexts, such as passages or documents by models on the KILT leaderboard, is found to be minimal, according to the investigation conducted by (Krishna et al., 2021). This lack of utilization poses a challenge for retrieval-augmented models aiming to enhance their performance in LFQA tasks. More specifically, the retrieved contexts fail to contribute significantly to acquiring new information necessary for generating comprehensive answers, impeding the models' progress.

One of the main reasons for the above problem is that previous models retrieve relevant contexts from a provided knowledge source or corpus, relying solely on the question itself. They overlooked the possibility that the question alone might not provide sufficient information to identify the relevant contexts within the corpus. Additionally, generating detailed long-form answers often entails aggregating knowledge snippets from diverse sources to provide a comprehensive explanation. To overcome these limitations, we propose a Long-Form Question Answering (LFQA) model with *iterative planning, retrieve, and generation* (**IPRG**) approach. The idea behind our model is that even if the question does not have enough information for a complete answer generation, we may use that as an initial query to construct a plan, retrieve contexts, and generate a preliminary answer. This preliminary answer can then provide new hints for gathering further information by the next step of planning and retrieving. Consequently, it will generate more detailed answers for the LFQA task.

To evaluate the performance of the proposed models, we conduct both quantitative and qualitative evaluations on two datasets from two different domains. As specified before the limitation of the ELI5 dataset, we use two datasets with more definitive long-form answers from both an open domain (i.e., wikiHow) and a specific technical domain (i.e., apple exchange dataset).

2 Methodology

The proposed IPRG model addresses the task of LFQA by taking a question q and a corpus $\mathcal{C}$. The corpus can be obtained through methods such as web search or by using a static corpus like Wikipedia. The model consists of three modules: 1) A Keyword Plan Generator $p(w_i|[q; y_{1:i-1}])$ that generates a keyword plan w_i consisting of a set of keywords for the next answer sentence given the concatenation of the question q and already generated answer sentences $y_{1:i-1}$, 2) A Retriever $p(C_i|[q; w_i])$ that retrieves top k passages as contexts C_i from $\mathcal{C}$ for supporting the next answer sentence generation based on the question q and current keyword plan w_i, and 3) an Answer Generator $p(y_i|[q; k_i; C_i])$ that generates the next sentence for answer given the question, keyword plan and retrieved passages. In the following subsections, we will describe each of these modules, followed by how they are combined and trained for the LFQA task. The overview of the proposed architecture is shown in Figure 1.

2.1 Keyword Plan Generator

A long-form answer demands detailed information. However, the conventional text generation models generally can generate short text with informativeness and coherence and tend to hallucinate or be repetitive if we force them to generate longer texts. One possible solution for this can be iteratively generating short text (i.e., sentence) at each iteration rather than generating the whole output at once. However, it still needs to solve the problem of generating repetitive sentences. To solve this issue, we predict some key points that will be used to guide the generation of the next answer sentence at each iteration. In other words, we plan to generate some future keywords that will be discussed in the next round of answer sentences. These keywords also help find relevant contexts from the given corpus in the subsequent step.

To do so, we frame this as a text-to-keyword generation (or prediction) problem, where, given a prompt, it generates a set of keywords. Initially, the prompt is the Question itself. In the subsequent iterations, the generated answers portions get concatenated with the Question to make the next prompt for generating the next set of keywords. More specifically, the Keyword Plan Generator $p(w_i|[q; y_{1:i-1}])$ generates w_i consisting of a set of keywords for the next answer sentence given the concatenation of the Question q and already generated answer sentences $y_{1:i-1}$.

To train the keyword plan generator, we convert each of the QA pairs in the training dataset into some texts to keyword-set pairs. For example, a question: "How to put or move downloaded files in different folders depending on file type?" and the first sentence of the ground truth answer is "I don't know of any Safari extension doing this but you may use Automator to create a folder action attached to your preferred download folder sorting files according to their extension or kind to various folders." To convert the above pair into a training sample of the keyword plan generator, we first use an existing keyword extraction model to extract important keywords from the first sentence of the ground truth answer, which is, for instance, Keys = ['preferred download folder', 'safari exten-

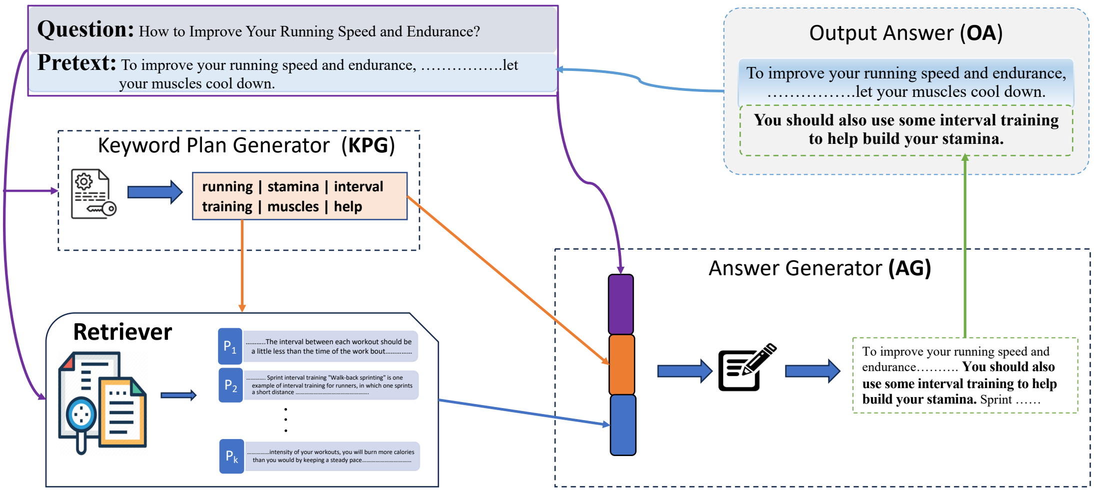

Figure 1: **IPRG**: At i-th iteration, **KPG** produces a set of keywords from the question and pretext. Then, **Retriever** retrieves k passages by using the question, pretext, and plan. Finally, the i-th sentence in the answer paragraph generated by **AG** is appended to **OA**. The **OA** serves as pretext at the next iteration.

sion', 'various folders', 'folder action', 'use']. We use these extracted Keys as ground truth keywords for the Question. So the task in training is, given the Question, can we generate the Keys? In other words, starting with the Question itself, we generate the keywords of the first answer sentence using a seq2seq model. Subsequently, we add each sentence of the answer one after another as input to predict the keywords of the next answer sentence. We initialized with pretrained BART (Lewis et al., 2019) for the text-to-keyword generation task.

2.2 Retriever

At each iteration, we construct a query by concatenating the original question, the previously generated answer (referred to as the pretext), and the generated keywords. This comprehensive query is then employed to retrieve relevant contexts, such as passages or sentences, from the existing corpus. To execute this retrieval process, we leverage an established Dense Passage Retrieval (DPR) model, as introduced by (Karpukhin et al., 2020). The goal is to extract the top k contexts from the corpus that are most pertinent to the query. This method ensures that our answer-generation process is enriched with relevant information, enhancing the overall coherence and informativeness of the responses.

2.3 Answer Generator

In this module, we combine the question, pretext, generated keywords, and retrieved contexts as inputs for our final answer generation model. This model is implemented as another sequence-to-sequence (seq2seq) architecture, initialized using BART (Lewis et al., 2019) as proposed by Lewis et al. (2019).

To address a potential issue of generating disconnected sentences and repetitive content in the answers, we explore an alternative approach. Instead of generating individual sentences, we opt to generate an entire paragraph in one sequence. This ensures coherence and reduces the likelihood of repeated information within the generated responses. Specifically, we generate a paragraph length and extract the first new sentence to append it to the final output.

During the training phase of this seq2seq model, we adopt a strategy to transform each original question-answer pair into multiple training samples. We initiate the process with only the question and progressively concatenate each subsequent sentence from the ground truth answer.

3 Experiments

WikiHowQA dataset: We prepare a novel long-form question-answering dataset, **WikiHowQA**, based on the WikiHow knowledge base [1]. Each article title is a **"How to"** question. Field experts have written these articles and provided a coherent paragraph summary for each one. Unlike summarization datasets (Cohen et al., 2021; Koupaee and Wang, 2018), we design this dataset for open-domain long-form question answering tasks, where the paragraph summaries of articles serve as the long-form answers to the title questions (as shown in Figure 2), and Wikipedia dump is used as knowl-

[1] https://www.wikihow.com/Main-Page

Models	WikiHow QA				Apple Exchange			
	Rouge-1		Rouge-L		Rouge-1		Rouge-L	
	Recall	F1	Recall	F1	Recall	F1	Recall	F1
GPT2	4.66	8.19	4.41	7.75	0.27	0.47	0.27	0.47
T5	4.02	6.68	3.78	6.26	0.30	0.49	0.30	0.49
Falcon	4.48	3.93	4.16	3.60	4.95	4.97	4.39	4.44
LLAMA2	28.60	21.87	26.21	19.95	**27.75**	18.57	**24.77**	16.50
LLAMA2 (w/o context)	33.26	25.38	30.30	22.99	**28.34**	19.40	**25.18**	17.19
BART	28.24	32.49	26.40	30.39	11.81	16.23	10.45	14.36
FiD	25.19	31.59	23.72	29.75	6.31	9.48	5.79	8.61
DPR + BART	28.71	32.74	26.78	30.54	12.88	17.22	11.32	15.16
MDR	33.50	33.30	31.01	30.40	22.74	21.50	20.76	19.56
IRG	33.68	3.29	31.30	30.99	23.88	21.78	21.70	19.75
IPRG	**35.36**	**33.65**	**32.88**	**31.25**	24.73	**22.13**	23.63	**20.42**

Table 1: Comparison with baselines (**Best** and 2nd best score)

edge corpus to retrieve relevant contexts. More details can be found in Appendix A.2.

Apple Exchange Dataset: To evaluate the proposed method in a technical domain, we use the Apple Exchange dataset adopted from a large COALA dataset (Rücklé et al., 2019). The answers to these technical questions require deeper technical knowledge compared to WikiHowQA. Thus, Wikipedia dumps are not sufficiently informative to answer these questions. We retrieve the top 10 sentences relevant to each ground truth answer sentence in Google search, excluding sentences from the StackExchange website, and compile them into a knowledge corpus. This corpus can be easily updated online. For comparison purposes, we create a static corpus by crawling the web search results.

3.1 Comparison with Baselines

Baselines: We compare our models with i) pretrained models such as GPT2-XL (Radford et al., 2019) and T5-3b (Raffel et al., 2020), LLAMA2 (Touvron et al., 2023) ii) Sequence-to-Sequence BART (Lewis et al., 2019) model, retrieval-augmented generation DPR+BART (Petroni et al., 2020) model, Fusion in Decoder (FiD) model (Izacard and Grave, 2020), MDR (Xiong et al., 2021). We fine-tune both BART, DPR+BART, FiD, and MDR on each target datasets and report results in Table 1. In the case of LLAMA2, we use the question and top 5 retrieved-context from DPR as a prompt to generate the long-form answer.

IRG: A variant of IPRG by excluding Keyword Plan Generator in which both Retriever and Answer Generator do not take any sequence of keywords as input. We implemented our model using hugging face [2] library functions and trained our models using the AdamW optimizer with the learning rate 2e-5.

Results: Both IPRG and IRG consistently outperform all the baselines in all metrics. Specifically, they outperform in recall scores (R-1 & R-L) by a large margin. In LFQA, the answers demand a larger coverage of ground truth information. In other words, the higher recall values imply more detailed answers with accurate information.

Planning enhances relevant Contexts Retrieval resulting in Comprehensive Answers. DPR+BART performs per BART, indicating that question-only retrieved contexts in a single pass have minimal impacts. On the other hand, iterative retrieval refines the contexts, resulting in the promising performance of the IRG model. Furthermore, the performance of the IPRG model is boosted by the two-fold advantages: i) keyword planning helps the retrieval module to identify and find out important information and the answer generator module to efficiently employ the current retrieved knowledge into answers at each iteration; ii) the understanding and knowledge about questions are getting enriched and refined by iterative characteristics of planning-retrieval-generation process. Thus, the performance improvement is reflected in both Rouge scores and entailment scores. Please refer to Section B.1 in the appendix for a detailed walkthrough of our whole process.

In the case of the Apple exchange dataset, we can see that LLAMA2 outperforms our model. The reason is that the quality of the corpus for this dataset is limited compared to that of WikiHow. And the LLAMA2 is not very dependent on the contexts while generating the answer, thus being less affected by the quality of the corpus. A more detailed analysis of our framework is shown in Appendix B.

4 Conclusion

This paper introduces a new approach for long-form question-answering tasks by iteratively planning content through keywords, retrieving contexts from a corpus, and generating answers using all the available information. Unlike existing long-form question-answering models that suffer from underutilization of retrieved contexts, our model demonstrates better retrieval of relevant contexts from diverse sources feeding the generated answer with iterative refinement. Experiments on multiple datasets show the superiority of our model over other state-of-the-art models.

[2] https://huggingface.co

5 Limitation

Firstly, the proposed model uses a keyword planning module that depends on existing keyword extraction methods for training the module. Therefore, the error from the keyword extraction method may propagate to the other modules, resulting in inconsistent and irrelevant answer generation. Developing a better task-dependent keyword planning generation that does not rely on the existing keyword extraction method can be a future research direction to improve the current model. Moreover, not all kinds of keywords are important as content plans for future answer generation. Therefore, automatically identifying informative keywords, in this case, can also improve the quality of the result.

Secondly, while training, the proposed IPRG model separately deals with each module, which also makes it susceptible to error propagation. There is no scope to learn from one module's error to refine the model to another. Therefore, developing a joint end-end model can be another future research direction in this regard.

We are aware of the capability of models such as GPT-4 or LLAMA, but we are unable to perform experiments by fintuing on them due to the limitation of budget and computational resources. However, given that more recent large language models have better text generation capabilities, using those in our proposed framework will also improve the performance of QA. As we used BART as our generative model, we use the BART or other contemporary LMs like GPT-2 and T5.

References

Nachshon Cohen, Oren Kalinsky, Yftah Ziser, and Alessandro Moschitti. 2021. WikiSum: Coherent summarization dataset for efficient human-evaluation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 212–219, Online. Association for Computational Linguistics.

Meri Coleman and Ta Lin Liau. 1975. A computer readability formula designed for machine scoring. *Journal of Applied Psychology*, 60(2):283.

Edgar Dale and Jeanne S Chall. 1948. A formula for predicting readability: Instructions. *Educational research bulletin*, pages 37–54.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. Eli5: Long form question answering. *arXiv preprint arXiv:1907.09190*.

James N Farr, James J Jenkins, and Donald G Paterson. 1951. Simplification of flesch reading ease formula. *Journal of applied psychology*, 35(5):333.

Michael Glass, Gaetano Rossiello, Md Faisal Mahbub Chowdhury, Ankita Naik, Pengshan Cai, and Alfio Gliozzo. 2022. Re2G: Retrieve, rerank, generate. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2701–2715, Seattle, United States. Association for Computational Linguistics.

Robert Gunning et al. 1952. Technique of clear writing.

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

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*, pages 874–880, Online. Association for Computational Linguistics.

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*.

Mahnaz Koupaee and William Yang Wang. 2018. Wikihow: A large scale text summarization dataset. *arXiv preprint arXiv:1810.09305*.

Kalpesh Krishna, Aurko Roy, and Mohit Iyyer. 2021. Hurdles to progress in long-form question answering. *arXiv preprint arXiv:2103.06332*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

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.

Yuning Mao, Pengcheng He, Xiaodong Liu, Yelong Shen, Jianfeng Gao, Jiawei Han, and Weizhu Chen. 2021. Rider: Reader-guided passage reranking for open-domain question answering. *arXiv preprint arXiv:2101.00294*.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. Ms marco: A human generated machine reading comprehension dataset. *choice*, 2640:660.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, et al. 2020. Kilt: a benchmark for knowledge intensive language tasks. *arXiv preprint arXiv:2009.02252*.

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, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. *arXiv preprint arXiv:1806.03822*.

Andreas Rücklé, Nafise Sadat Moosavi, and Iryna Gurevych. 2019. Coala: A neural coverage-based approach for long answer selection with small data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6932–6939.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*.

RJ Senter and Edgar A Smith. 1967. Automated readability index. Technical report, Cincinnati Univ OH.

Dan Su, Xiaoguang Li, Jindi Zhang, Lifeng Shang, Xin Jiang, Qun Liu, and Pascale Fung. 2022. Read before generate! faithful long form question answering with machine reading. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 744–756, Dublin, Ireland. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Siyuan Wang, Zhongyu Wei, Zhihao Fan, Qi Zhang, and Xuanjing Huang. 2022. Locate then ask: Interpretable stepwise reasoning for multi-hop question answering. *arXiv preprint arXiv:2208.10297*.

Wenhan Xiong, Xiang Li, Srini Iyer, Jingfei Du, Patrick Lewis, William Yang Wang, Yashar Mehdad, Scott Yih, Sebastian Riedel, Douwe Kiela, and Barlas Oguz. 2021. Answering complex open-domain questions with multi-hop dense retrieval. In *International Conference on Learning Representations*.

Wenhan Xiong, Xiang Lorraine Li, Srini Iyer, Jingfei Du, Patrick Lewis, William Yang Wang, Yashar Mehdad, Wen-tau Yih, Sebastian Riedel, Douwe Kiela, et al. 2020. Answering complex open-domain questions with multi-hop dense retrieval. *arXiv preprint arXiv:2009.12756*.

Semih Yavuz, Kazuma Hashimoto, Yingbo Zhou, Nitish Shirish Keskar, and Caiming Xiong. 2022. Modeling multi-hop question answering as single sequence prediction. *arXiv preprint arXiv:2205.09226*.

A Appendix

A.1 Related Works

Open domain question answering is the task of answering questions by utilizing a knowledge base or corpus. In this setup, passage retrieval is a key step to retrieving supporting documents before answering the question. A plethora of research has been conducted on retrieval-augmented question answering (Lewis et al., 2020; Petroni et al., 2020; Mao et al., 2021; Izacard and Grave, 2021; Nguyen et al., 2016). Existing works mostly focus on answering factoid questions.

However, due to the recent emergent of generative models (Radford et al., 2019; Lewis et al., 2019; Raffel et al., 2020), long-form question answering has become an active research area crowded by lots of unavoidable challenges such as factual hallucination, retrieving relevant context, and fluent and logically consistent answer generation. RAG (Lewis et al., 2020), an end-to-end retrieval-based generative model, which incorporates DPR (Karpukhin et al., 2020) to retrieve supporting passages and then employ BART (Lewis et al., 2019) to generate long-form answers from questions concatenated with evidence documents. Instead of concatenating document texts, FID (Izacard and Grave, 2021) encodes first retrieved documents independently and then fuses representations into a decoder to generate answers. RBG (Su et al., 2022) combines outputs from fusion-in-decoder module and machine reading comprehension module by following the mechanism of pointer-generator network (See et al., 2017). To improve the contexts, Re^{2}G (Glass et al., 2022) augment RAG architecture by adding a learnable reranker module to select top k passages from a pool of passages from multiple retrievers. It is to be noted that all of these models perform their retrieval process once using questions as queries and generate answers in a single hop.

Recently, several multi-hop question-answering models have been introduced in (Xiong et al., 2020; Wang et al., 2022; Yavuz et al., 2022), but are limited to factoid question answering.

A.2 WikiHowQA dataset

We collect article titles that have coherent paragraph summaries by filtering out those which have no paragraph-styled summaries or incomplete summaries. Consequently, the WikiHowQA dataset comprises 37,815 question-answer pairs. One of the key attributes of this dataset is that the answers are expert-written in plain English, which requires less world knowledge to understand and evaluate the answers. In order to measure the readability scores, we use classical readability metrics such as FKGL (Farr et al., 1951), GFI (Gunning et al., 1952), ARI (Senter and Smith, 1967), CLI (Coleman and Liau, 1975), DCR (Dale and Chall, 1948). All metrics' generated scores indicate the number of years of formal education required for a native English speaker to understand the answer text. As reported in Table 2, the scores for WikiHowQA answers are significantly smaller than that of the widely used ELI5 dataset.

Datasets	Readability				
	FKGL	DCR	ARI	CLI	GFI
WikiHowQA	**8.70**	**7.78**	**9.12**	**7.58**	**11.37**
ELI5	9.81	8.38	10.22	8.97	12.70

Table 2: Readability scores for reference answers. Smaller value is more Readable.

B Result Analysis

IPRG generates more Entailed less Contradictory answer. Not only comparing performances by measuring the text overlapping with references, but we also compute entailment scores to evaluate how much the model's generated answers are logically aligned to the ground truth. We leverage a pretrained BART-large-mnli model (available on Huggingface[3]) where the generated answers are considered as hypothesis and ground-truth as the NLI premise. This model calculates the scores indicating the logical entailment, contradiction, or neutrality of a hypothesis with respect to the premise. As reported in Table 3, IPRG achieves more entailment scores as well as less contradictory scores in both datasets. Whereas, DPR+BART generated answers contain more contradiction. This happens because the contexts retrieved using only questions might lack relevant important information.

Models	WikiHow QA		Apple Exchange	
	NLI		NLI	
	Entailed	Contradict	Entailed	Contradict
Falcon	38.61	15.23	48.70	17.01
LLAMA2 (w/o context)	**10.35**	3.31	7.25	3.67
LLAMA2	**9.79**	**3.30**	7.74	**3.32**
DPR + BART	7.29	20.37	<u>10.00</u>	20.49
MDR	6.07	12.48	9.63	20.77
IRG	7.09	14.86	9.52	19.56
IPRG	<u>8.34</u>	<u>10.98</u>	**12.74**	<u>13.88</u>

Table 3: Factual Consistency Comparison

A Qualitative Case Study: Our IPRG model can effectively capture relevant aspects to generate an elaborative answer for the target question. For instance, in Figure 2, the ground-truth answer mentions three ways to improve through: i) <u>Strength training</u>, ii) **Interval training**, and iii) *Stretching*. A well-detailed answer should include all aspects. We can see that both IPRG and IRG variants showed superior performance in capturing aspects. DPR+BART only includes details about <u>strength</u> training but failed to capture the other two. On the other hand, IPRG perfectly captures all three aspects while *stretching* aspect is absent in IRG's answer.

B.1 Model Workflow with an Example

In this section, we illustrate our model workflow using an example step-by-step iteratively shown in Figure 4 and 5. Each row of these tables contains the question, current pretext, and generated keywords at each iteration step in the first column. For convenience, we display the top 2 retrieved passages in the second column.

[3] https://huggingface.co/facebook/bart-large-mnli

Question: How to Improve Your Running Speed and Endurance?

Ground Truth

To improve your running speed and endurance, start by doing 10 walking lunges for each leg to stretch out your muscles and reduce the risk of cramps and injuries while you run. **Additionally, add interval training by running for 1 minute at maximum speed to raise your heart rate and exhaust your muscles. Then, walk for 2 minutes to let your muscles cool down.** Repeat this process 4 times in the middle of your run, but don't do it for more than 2 runs in a 10-day period to avoid hurting yourself.

IPRG Model

To improve your running speed and endurance, perform a "stretch"—walking lunges, or a "run-stretch" where you stand up straight and push yourself up to avoid cramps. It involves stretching your arms and legs over your body. Start by sprinting for 5 to 10 minutes to build up your endurance, then take a short break and repeat the process 4 to 5 times to warm up your muscles. **Repeat these exercises 4 times in the middle of your run, which will give your muscles time to rest and recover. After your run is over, rest for 2 to 3 minutes to let your muscles cool down. You should also use some interval training to help build your stamina. sprint for 10 seconds on a hard surface, then 5 minutes on a softer surface to work your core muscles.**

IRG

To improve your running speed and endurance, start by sprinting for 10 to 15 se conds, which will give your body time to rest and cool down. Then, repeat the process 4 to 5 times, alternating which times you sprint. Repeat these intervals 4 times, bouncing between each time you start running. After you've built up al l of your endurance, run for 2 to 3 minutes to warm up your muscles and give t hem a chance to rest. Finally, repeat these stretches 4 times in the middle of yo ur run to give your muscles a rest.

MDR

To improve your running speed and endurance, start by sprinting for 10 to 15 seconds, which will give your body time to rest and cool down. Then, repeat the process 4 to 5 times, alternating which times you sprint. Repeat these intervals 4 times, bouncing between each time you start running. *After you've built up all of your endurance, run for 2 to 3 minutes to warm up your muscles and give them a chance to rest. Finally, repeat these stretches 4 times in the middle of your run to give your muscles a rest.*

LLAMA2

Improving your running speed and endurance requires a well-structured training program that includes various components such as interval training, tempo runs, hill repeats, and long runs. Interval training involves alternating between periods of high-intensity running and active recovery, typically done at race pace or faster. Tempo runs involve sustained runs at a slower pace than race pace, usually around 5K to 10K race pace. Hill repeats involve running up a steep hill at a high intensity, followed by active recovery down the hill. Long runs are essential for building endurance and should be done at a steady state without any interruptions. It is important to vary the intensities and distances of your runs to accommodate different physiological adaptations and to avoid plateaus. Additionally, incorporating strength training and cross-training activities can help improve overall running performance.

FiD

To improve your running speed and endurance, start by choosing a running route that you can run at your own pace. Then, choose a route that you can run at your own pace, and make sure you follow a specific route. Next, set a goal of running for a certain amount of time each week, and run for that number of days each week. Additionally, try to run for at least 30 minutes a day, and then gradually increase the distance you run to the next level.

Figure 2: Comprehensiveness of answers w.r.t. three aspects: Strengthening, **Interval** training, *Stretching*.

Query	Retrieved top two Passages
Question: How to Improve Your Running Speed and Endurance? **Pretext**: "" **Keywords**: walking lunges, run, endurance, stretch, cramps	**P1**: including 2–3 days of core strengthening. In the base phase the athlete now works on building cardiovascular **endurance** by having several long **runs** staying in heart rate zone 1-2 every week and each week adding slightly more mileage (using 10% rule for safely increasing the mileage). Core strengthening is continued in the base period. Once the base phase is complete and the athlete has sufficient **endurance**, the build period is needed to give the athlete the ability to hold a faster pace for the race duration. The build phase is where duration of runs is traded for intensity. **P2**: **Stretching** does not appear to reduce the risk of injury during exercises, except perhaps a dynamic warm-up for runners. While running places extreme stress loads on the joints, static **stretching** can help to improve joint flexibility. However, this has not been proven to reduce risk of injury in the runners. A dynamic (stretching) warm up has been shown to help overall running performance. One should avoid overtraining to prevent **cramps**. The calf muscles slowly respond to
Question: How to Improve Your Running Speed and Endurance? **Pretext**: To improve your **running** speed and **endurance**, perform a "**stretch**"—**walking lunges**, or a "run-stretch" where you stand up straight and push yourself up to avoid **cramps**. **Keywords**: stretch, arms, muscles	**P1**: leg muscles. Unlike the previous exercises, these are outdoor activities. The calf exercises should be performed every 3–4 days. They can be alternated and interchanged to avoid getting used to the load. General Workout Tips Before any serious activity, including doing calf-building exercises, the muscles and joints should be properly warmed up. Aerobic exercises should be performed at the beginning of a workout. A workout should end with calf strengthening exercises to stimulate their growth, and stretching that involves taking a wide step backward, placing the heel on the floor, and bending the torso forward. Each leg should be stretched for 10–20. **P2**: must return to, and pause in, the correct starting position before continuing. If you rest on the ground or raise either hand or foot from the ground, your performance will be terminated. You may reposition your hands and/or feet during the event as long as they remain in contact with the ground at all times. Correct performance is important. You will have two minutes in which to do as many push-ups as you can." Sit-up "The sit-up event measures the endurance of the abdominal and hip-flexor muscles. On the command 'get set,' assume the starting position by lying on your.
Question: How to Improve Your Running Speed and Endurance? **Pretext**: To improve your running speed and endurance, perform a "stretch"—walking lunges, or a "run-stretch" where you stand up straight and push yourself up to avoid cramps. It involves **stretching** your **arms** and legs over your body. **Keywords**: sprinting, repeat, process, minutes, times	**P1**: in losing weight, staying in shape and improving body composition. Research suggests that the person of average weight will burn approximately 100 calories per mile run. Running increases one's metabolism, even after running; one will continue to burn an increased level of calories for a short time after the run. Different speeds and distances are appropriate for different individual health and fitness levels. For new runners, it takes time to get into shape. The key is consistency and a slow increase in speed and distance. While running, it is best to pay attention to how one's body feels. **P2**: Many training programs last a minimum of five or six months, with a gradual increase in the distance run and finally, for recovery, a period of tapering in the one to three weeks preceding the race. For beginners wishing to merely finish a marathon, a minimum of four months of running four days a week is recommended. Many trainers recommend a weekly increase in mileage of no more than 10%.
Question: How to Improve Your Running Speed and Endurance? **Pretext**: To improve your running speed and endurance, perform a "stretch"—walking lunges, or a "run-stretch" where you stand up straight and push yourself up to avoid cramps. It involves stretching your arms and legs over your body. Start by **sprinting** for 5 to 10 **minutes** to build up your endurance, then take a short break and **repeat** the **process** 4 to 5 **times** to warm up your muscles. **Keywords**: run, repeat, muscles, reps, recover	**P1**: a base for more intense workouts by strengthening the heart and increasing the muscles' ability to use oxygen, and to recover between hard workouts. Daniels recommends that most training miles are performed in E pace. Typical E runs include continuous runs up to about an hour. Marathon (M) pace At 80-85% HR_{max}, this intensity is primarily aimed towards runners training for the marathon. The pace is one at which the runner hopes to compete. The pace can be included in other programs for a more intense workout, especially if the runner feels fresh and there is enough time to recover afterwards. **P2**: in losing weight, staying in shape and improving body composition. Research suggests that the person of average weight will burn approximately 100 calories per mile run. Running increases one's metabolism, even after running; one will continue to burn an increased level of calories for a short time after the run. Different speeds and distances are appropriate for different individual health and fitness levels. For new runners, it takes time to get into shape. The key is consistency and a slow increase in speed and distance. While running, it is best to pay attention to how one's body feels.
Question: How to Improve Your Running Speed and Endurance? **Pretext**: To improve your running speed and endurance, perform a "stretch"—walking lunges, or a "run-stretch" where you stand up straight and push yourself up to avoid cramps. It involves stretching your arms and legs over your body. Start by sprinting for 5 to 10 minutes to build up your endurance, then take a short break and repeat the process 4 to 5 times to warm up your muscles. **Repeat** these exercises 4 times in the middle of your **run**, which will give your **muscles** time to rest and **recover**. **Keywords**: stretch, arms, legs, run, repeat, times	**P1**: must return to, and pause in, the correct starting position before continuing. If you rest on the ground or raise either hand or foot from the ground, your performance will be terminated. You may reposition your hands and/or feet during the event as long as they remain in contact with the ground at all times. Correct performance is important. You will have two minutes in which to do as many push-ups as you can." Sit-up "The sit-up event measures the endurance of the abdominal and hip-flexor muscles. On the command 'get set,' assume the starting position by lying on your **P2**: in losing weight, staying in shape and improving body composition. Research suggests that the person of average weight will burn approximately 100 calories per mile run. Running increases one's metabolism, even after running; one will continue to burn an increased level of calories for a short time after the run. Different speeds and distances are appropriate for different individual health and fitness levels. For new runners, it takes time to get into shape. The key is consistency and a slow increase in speed and distance. While running, it is best to pay attention to how one's body feels.

Table 4: Walk through IPRG model workflow with an example

Query	Retrieved top two Passages
Question: How to Improve Your Running Speed and Endurance? **Pretext**: To improve your running speed and endurance, perform a "stretch"—walking lunges, or a "run-stretch" where you stand up straight and push yourself up to avoid cramps. It involves stretching your arms and legs over your body. Start by sprinting for 5 to 10 minutes to build up your endurance, then take a short break and repeat the process 4 to 5 times to warm up your muscles. Repeat these exercises 4 times in the middle of your run, which will give your muscles time to rest and recover. After your run is over, run for 2 to 3 minutes to let your muscles cool down. **Keywords**: running, stamina, interval training, muscles, help	**P1**: cope with the intensity, and to train for longer periods of time, this training is performed as interval training, hence the name. The interval between each workout should be a little less than the time of the work bout. Optimum intervals are 3–5 minutes long. There is no benefit to exceeding 5 minutes at this pace, under Daniels' theory, which means that despite the popularity of mile-repeats in many running groups, Daniels discourages them for people whose pace is slower than about 5:00/mile, preferring shorter intervals such as 1200 meters. **P2**: (a minimum of 45 minutes). The development of aerobic and anaerobic capacities, and the adaptability of fartlek - to mimic running during specific sports - are characteristics it shares with other types of interval training. Sprint interval training "Walk-back sprinting" is one example of interval training for runners, in which one sprints a short distance (anywhere from 100 to 800 metres), then walks back to the starting point (the recovery period), to repeat the sprint a certain number of times. To add challenge to the workout, each of these sprints may start at predetermined time intervals - e.g. 200 metre. One should avoid overtraining to prevent cramps. The calf muscles slowly respond to
Question: How to Improve Your Running Speed and Endurance? **Pretext**: To improve your running speed and endurance, perform a "stretch"—walking lunges, or a "run-stretch" where you stand up straight and push yourself up to avoid cramps. It involves stretching your arms and legs over your body. Start by sprinting for 5 to 10 minutes to build up your endurance, then take a short break and repeat the process 4 to 5 times to warm up your muscles. Repeat these exercises 4 times in the middle of your run, which will give your muscles time to rest and recover. After your run is over, run for 2 to 3 minutes to let your muscles cool down. You should also use some **interval training** to help build your **stamina**. **Keywords**: run, sprint, muscles, time	**P1**: workouts the day after interval sessions. Finally the race phase of the periodization approach is where the duration of the workouts decreases but intense workouts remain so as to keep the high lactate threshold that was gained in the build phase. In Ironman training, the race phase is where a long "taper" occurs of up to 4 weeks for highly trained Ironman racers. A final phase is designated transition and is a period of time, where the body is allowed to recover from the hard race effort and some maintenance endurance training is performed so the high fitness level. **P2**:(a minimum of 45 minutes). The development of aerobic and anaerobic capacities, and the adaptability of fartlek - to mimic running during specific sports - are characteristics it shares with other types of interval training. Sprint interval training "Walk-back sprinting" is one example of interval training for runners, in which one sprints a short distance (anywhere from 100 to 800 metres), then walks back to the starting point (the recovery period), to repeat the sprint a certain number of times. To add challenge to the workout, each of these sprints may start at predetermined time intervals - e.g. 200 metre.
Question: How to Improve Your Running Speed and Endurance? **Answer**: To improve your running speed and endurance, perform a "stretch"—walking lunges, or a "run-stretch" where you stand up straight and push yourself up to avoid cramps. It involves stretching your arms and legs over your body. Start by sprinting for 5 to 10 minutes to build up your endurance, then take a short break and repeat the process 4 to 5 times to warm up your muscles. Repeat these exercises 4 times in the middle of your run, which will give your muscles time to rest and recover. After your run is over, run for 2 to 3 minutes to let your muscles cool down. You should also use some interval training to help build your stamina. **Sprint** for 10 seconds on a hard surface, then 5 minutes on a softer surface to work your core **muscles**.	

Table 5: Walk through IPRG model workflow with an example (as the continuation of Table 4)