FlexiQA: Leveraging LLM's Evaluation Capabilities for Flexible Knowledge Selection in Open-domain Question Answering

Yuhan Chen[1]*, **Shuqili** [1]*, **Rui Yan**[1]†
[1]Gaoling School of Artificial Intelligence, Renmin University of China
{yuhanchen, shuqili, ruiyan}@ruc.edu.cn

Abstract

Nowadays, large language models (LLMs) have demonstrated their ability to be a powerful knowledge generator of *generate-then-read* paradigm for open-domain question answering (ODQA). However this new paradigm mainly suffers from the "hallucination" and struggles to handle time-sensitive issue because of its expensive knowledge update costs. On the other hand, *retrieve-then-read*, as a traditional paradigm, is more limited by the relevance of acquired knowledge to the given question. In order to combine the strengths of both paradigms, and overcome their respective shortcomings, we design a new pipeline called "FlexiQA", in which we utilize the diverse evaluation capabilities of LLMs to select knowledge effectively and flexibly. First, given a question, we prompt an LLM as a discriminator to identify whether it is time-sensitive. For time-sensitive questions, we follow the *retrieve-then-read* paradigm to obtain the answer. For the non-time-sensitive questions, we further prompt the LLM as an evaluator to select a better document from two perspectives: factuality and relevance. Based on the selected document, we leverage a reader to get the final answer. We conduct extensive experiments on three widely-used ODQA benchmarks, the experimental results fully confirm the effectiveness of our approach. Our code and datasets are open at https://github.com/Fiorina1212/FlexiQA

1 Introduction

Open-domain question answering (ODQA) as a knowledge-intensive task, necessitate a substantial amount of world knowledge to be effective (Petroni et al., 2020). Current methods for handling ODQA often share two common paradigms: the *retrieve-then-read* paradigm, which consists of retrieving a small set of relevant contextual documents from sources, and then generating the answer on both the question and the retrieved documents (Karpukhin et al., 2020; Lewis et al., 2020; Izacard and Grave, 2020); and the *generate-then-read* paradigm, which initiates by prompting an LLM to generate contextual documents based on the question, then by reading and extracting relevant information from the generated documents to generate the final answer. Nevertheless, these two type of paradigms are with their own drawbacks.

For the *retrieve-then-read* paradigm, candidate documents are chunked and fixed for a given question. Moreover, the frequently-used two-tower dense retrieval models (Karpukhin et al., 2020) often leads to superficial interactions between the document and the question (Khattab et al., 2021). These can result in some retrieved documents containing irrelevant or noisy data that is not pertinent to the question. For the *generate-then-read* paradigm, though there are works show that the generated contextual documents contain the correct answer more often than the top retrieved documents (Yu et al., 2022), there are still some imperative issues to be solved. LLMs are hard to expand or revise their memory since all the information needs to be stored in the parameters (Geva et al., 2021). Moreover, they can't straightforwardly provide insight into their generations, and may produce "hallucinations" (Lewis et al., 2020; Lv et al., 2023c) or struggle to address time-sensitive issue. A time-sensitive question is one whose answer will change over time. For example, *Where will the next Olympic Games be held?* is time-sensitive, while *Who wrote the book 'The Razor's Edge'?* is not time-sensitive. Time-sensitivity becomes a non-negligible issue when leveraging LLMs for ODQA.

*Equal contribution.
†Corresponding author: Rui Yan (ruiyan@ruc.edu.cn).

Findings of the Association for Computational Linguistics: EACL 2024, pages 56–66
March 17-22, 2024

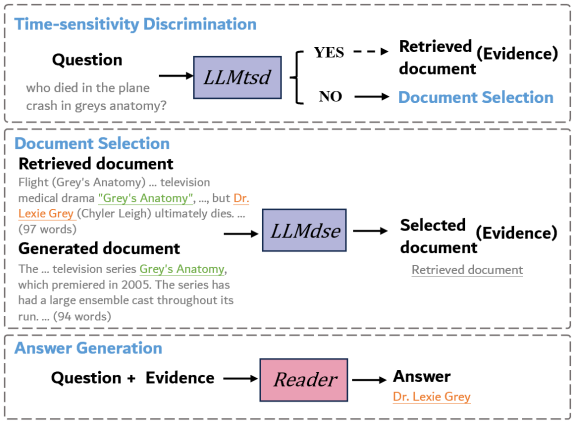

Figure 1: Overview of FlexiQA, including three parts: time-sensitivity discrimination, document selection and answer generation. Besides, an example is shown in gray color.

There are a few works analyzed it recently (Yu et al., 2022; Zhang and Choi, 2021), but didn't try to solve it. Meanwhile, retrieval-based models have no such problem because it is easy to replace the external knowledge source and access the time-aligned documents.

Based on the aforementioned observations, we unify these two paradigms and proposed a new ODQA pipeline called FlexiQA. Overall, our contributions are listed as follows:

• We propose FlexiQA as a unified pipeline which flexibly leverages the multi-dimensional evaluation ability of LLMs for ODQA for the first time. By evaluating the question and the documents obtained by retriever and generator from multi-perspective, the better one is picked to enhance the answer generation. FlexiQA could tackle three drawbacks of the two classic paradigms: the time-sensitive issue, the irrelevance issue and the non-factuality issue.

• We tackle the time-sensitive issue of LLMs for the first time in ODQA task. We prompt an LLM to discriminate if the given question is time-sensitive or not. Then we design different answering strategy for different type question. Moreover, we release two time-sensitivity annotated datasets for widely research on this issue in the future.

• We conduct extensive experiments for ODQA task on three benchmarks, and FlexiQA achieves the new state-of-the-art performance.

2 Related Work

2.1 Open-Domain Question Answering

Open-domain generation poses a longstanding challenge (Lv et al., 2023a,b) in the field of natural language processing. Within this realm, Open-Domain Question Answering (ODQA) stands out as one of the most extensively studied tasks. It has garnered significant attention from both industry and academia in recent years (Liu et al., 2022). Up to now, most recent works are built following the two basic paradigms, *retrieve-then-read* and *generate-then-read*.

Retrieve-Then-Read Paradigm The retriever first retrieve evidence documents based on the given question from a large external corpus. Then the reader intends to generate answer condition on both the evidence and the given question. Many recent works focus on improving the retriever (Khattab et al., 2021; Qu et al., 2020). The readers based on PLMs such as T5 (Raffel et al., 2020) and InstructGPT (Ouyang et al., 2022) have become a common choice with the develop of LLMs (Izacard and Grave, 2020; Cheng et al., 2021; Yu et al., 2022; Chen et al., 2023).

Generate-Then-Read Paradigm Many works have demonstrated that the knowledge stored in the parameters of LLMs could serve as a "retriever" to some extent by directly generating text (Petroni et al., 2019; Roberts et al., 2020). Based on that, Yu et al. (2022) exploit the potential of directly generating contextual documents for open-domain questions and propose the *generate-then-read* paradigm. This paradigm directly generates contextual documents for a given question instead of retrieving documents from an external corpus.

2.2 Evaluation Ability of LLMs

Recently, utilizing LLMs as evaluators becomes a natural idea for their remarkable performance across various tasks (Kushman et al., 2014; Roy and Roth, 2016; Bubeck et al., 2023). LLMs aligned with Reinforcement Learning from Human Feedback (RLHF, Ouyang et al., 2022; Wang et al., 2022) are used to evaluate and compare the generations from different models. Other works prompt LLMs to achieve self-verify, self-refine, and self-debug ability in zero-shot setting (Shinn et al., 2023; Weng et al., 2022; Madaan et al., 2023). Especially, vicuna's evaluation pipeline (Chiang et al., 2023) has obtained significant interest, which

leverages GPT-4 to score and compare candidate responses and provide explanations.

In our work, we unify these two paradigms into a new pipeline and leverage the evaluation ability of LLMs to enhance the ODQA performance for the first time.

3 Method of FlexiQA

Under the zero-shot setting, we will introduce the details of our proposed pipeline as shown in Figure 1 which comprises three steps: *Time-sensitivity Discrimination, Document Selection, Answer Generation*. First, we prompt an LLM to discriminate if the given question is time-sensitive. If the answer is **YES**, we choose the retrieved document as the evidence. Otherwise, we further prompt the LLM as an evaluator to decide which document (one is from generation, another one is from retrieval) is better from two perspectives: factuality and relevance. And finally we use the picked document as evidence to answer the given question by a reader.

3.1 Time-Sensitivity Discrimination

In this subsection, we design an evaluation prompt template for time-sensitivity discrimination with one placeholder Q: $T_{ts}(Q)$. Given a question Q, a prompt $T_{ts}(Q)$ is produced by the designed template. Then we instruct an LLM with $T_{ts}(Q)$ to determine whether the given question Q is time-sensitive and LLM will give feedback to us with a the $Label_{ts}$ = **YES**/**NO**. The role of LLM here is a time-sensitivity discriminator, named $LLM_{tsd}(\cdot)$. Formally, we describe this process with the following formula: $Label_{ts} = LLM_{tsd}(T_{ts}(Q))$. The details of the prompt template is described in Appendix B.

As mentioned in Introduction, retrieval-based models won't severely affected by time-sensitive issue because it is easy to replace the external knowledge source and then access the time-aligned documents. For the questions with $Label_{ts}$ = **YES** (i.e. the question is time-sensitive), we directly employ Information Retrieval (IR) to obtain the final evidence document: $E = IR(Q)$. For the non-time-sensitive questions, we obtain both the generated document from an LLM generator $LLM_{kg}(\cdot)$ and the retrieved document from a retriever IR: $G_{doc} = LLM_{kg}(Q), R_{doc} = IR(Q)$.

3.2 Document Selection

Now for the non-time-sensitive questions, inspired by the multi-dimensional evaluation ability of LLMs, we leverage it here to unify the *generate-then-read* paradigm and the *retrieve-then-read* paradigm. Specifically, we leverage LLMs to compare two documents from two perspective, the factuality and relevance, then pick the better one as the evidence.

We design another evaluation template $T_{ds}(Q, G_{doc}, R_{doc})$ for document selection, which includes three placeholders for the given question Q, the generated document G_{doc} and the retrieved document R_{doc}. See Appendix B for the detail description of evaluation template.

For any question, a prompt according to this template is produced and is used to instruct an LLM to score the two given documents. Next, the LLM output the document with higher overall score to serve as the evidence. The role of this LLM is a document selection evaluator, named $LLM_{dse}(\cdot)$. Formally, we describe this process with the following formula: $E = LLM_{dse}(T_{ds}(Q, G_{doc}, R_{doc}))$.

3.3 Answer Generation

After the two steps above, we obtain the optimal evidence corresponding to the given question, which draw upon the two classic paradigms' strong points and make up the shortcomings. Combining the question Q and the evidence E, we utilize another LLM as a reader $LLM_{reader}(\cdot)$ to get the final answer: $Answer = LLM_{reader}(Q, E)$.

4 Experiments

4.1 Datasets & Metrics

We conduct comprehensive experiments on three widely used benchmarks: NaturalQuestions (NQ, Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), WebQuestions (WebQ, Berant et al., 2013). More detailed information can be found in Table 3 in Appendix A. We use Exact Match (EM) score (Zhu et al., 2021) and F1 score to evaluate models' performance since the correct answer is not an flexible and open answer. For EM score, an answer is considered correct if and only if its normalized form has a match in the acceptable answer list. F1 score measures the recall of answer at the token level.

4.2 Baselines

We compare our pipeline with the following strong baselines. (1) BM25 (Robertson et al., 1995) + InstructGPT; (2) Contriever (Izacard et al., 2022) + InstructGPT; (3) Google + InstructGPT; (4)

DPR (Karpukhin et al., 2020) + InstructGPT; (5) InstructGPT (no docs.) (Ouyang et al., 2022); (6) GENREAD (Yu et al., 2022); (7) Vanilla-United: To fully evaluate the effectiveness of our proposed method, we also compare our pipeline with another vanilla method which combines the two documents from retrieval and generation as evidence directly without comparison. See Appendix A.2 for the details of above baselines.

4.3 Implementation Details

We follow the experimental settings as in GENREAD, and utilize *text-davinci-002* version of InstructGPT (Ouyang et al., 2022) for the knowledge generator LLM_{kg} and the reader LLM_{reader}. We employ *dpr-multi* version of DPR (Karpukhin et al., 2020) as the retriever. We leverage the *gpt-3.5-turbo* as discriminators LLM_{tsd} and LLM_{dse}. The generation temperature is set to $T = 0$ to ensure the reproducibility.

4.4 Results

As shown in Table 1, our approach surpasses all previous methods and achieves the state-of-the-art performance with improvements of 3.3, 1.2, and 0.3 points of EM score on NQ, TriviaQA, WebQ, respectively. The results demonstrate that our pipeline could select suitable knowledge sources effectively to enhance the ODQA performance. Moreover, Vanilla-United, as the simplest way to fuse two paradigm knowledge, yields worse results than FlexiQA. The part of reason for this result is that there are content conflicts between the generated document and the retrieved document partly due to the three issues mentioned above. We provide a more detailed results in Table 4 in Appendix C including F1 metric.

Models	NQ	TriviaQA	WebQ
*with retriever			
BM25+InstructGPT	19.7	52.2	15.8
Contriever+InstructGPT	18	51.3	16.6
Google+InstructGPT	28.8	58.8	20.4
DPR+InstructGPT	<u>29.1</u>	55.7	21.5
*without retriever			
InstructGPT (no docs.)	20.9	57.5	18.6
GENREAD	28.2	59	<u>24.8</u>
*with retriever and generator			
Vanilla-United	28.1	<u>59.3</u>	20.9
FlexiQA	**32.4**	**60.5**	**25.1**

Table 1: Exact match (EM) score on NQ, TriviaQA and WebQ test sets.The best performance model is in **bold** and the second one is in <u>underline</u>.

Models	NQ			WebQ		
	TS	non-TS	Total	TS	non-TS	Total
DPR+InstructGPT	**22**	<u>30.3</u>	<u>29.1</u>	**14.1**	21.6	21.5
GENREAD	17.6	29.7	28.2	9.9	<u>25.2</u>	<u>24.8</u>
Vanilla-United	17	29.6	28.1	9.9	21.4	20.9
FlexiQA	<u>21.9</u>	**33.6**	**32.4**	<u>11.3</u>	**25.6**	**25.1**

Table 2: The experiment results of time-sensitive issue. TS means the time-sensitive subset of NQ and WebQ, while non-TS means the non-time-sensitive subset.

4.5 Analysis

4.5.1 Analysis of Time-Sensitivity

To analyze the experiment results for time sensitivity, we annotated the time-sensitive label for NQ and WebQ test sets. Specifically, for every question in the dataset, we label it with time-sensitive (**YES**) or non-time-sensitive (**NO**). We release these two annotated dataset for widely research on this issue for the future works.

We compare the performance of our FlexiQA with representative baselines, DPR + InstructGPT of *retrieve-then-read* paradigm, GENREAD of *generate-then-read* paradigm, the naive unify method Vanilla-United, on both time-sensitive (TS) and non-sensitive (non-TS) subsets of two datasets. The experiment results are presented in Table 2. It can be seen that the retrieval-based method DPR + InstructGPT outperforms the generation-based method GENREAD by a significant margin on TS subset of both datasets, which confirms our motivation that *retrieve-then-read* paradigm could handle time-sensitive issue by nature.

The results indicate that our pipeline indeed has the ability to recognize time-sensitive questions and to tackle this issue, resulting in a improvement of 4.3 points and 1.4 points of EM score on the TS subsets comparing to *generate-then-read* method GENREAD. However, there is still a gap between FlexiQA and DPR + InstructGPT on the TS subsets, which can be attributed to the unsatisfactory zero-shot evaluation ability of LLMs for time-sensitive discrimination. This could be a key study object in the future. We provide a more detailed results in Table 5 in Appendix D including F1 metric.

4.5.2 Case Study of Document Selection

From the results on the non-TS subsets shown in Table 2, we can observe that FlexiQA is able to effectively select superior documents based on the evaluation of factuality and relevance. For both subsets, our FlexiQA has reached the optimal results

compared to other baselines. To further analyze the effectiveness of FlexiQA in document selection, we present three representative cases of three issues respectively in Appendix D. All the results show the strong performance of our FlexiQA.

5 Conclusion

In this paper, we unify two classic ODQA paradigms and propose a new pipeline called FlexiQA. FlexiQA leverages the multi-dimensional evaluation ability of LLMs flexibly for ODQA for the first time, and it tackles three existing drawbacks in the two classic paradigms: the time-sensitive issue, the irrelevance issue and the non-factuality issue. Moreover, we release two time-sensitivity annotated datasets for widely research on this issue in the future. Experimental evaluations show that our model achieves the best performance on three datasets.

Limitations

The limitations of our pipeline FlexiQA are stated briefly as follows:

- First, due to the setting of our study (in the context of large-scale zero-shot models), the influence of biases in large language models is inevitable. In practical applications, the efficient few-shot learning (Zhang et al., 2024) could enhance the overall effectiveness of the pipeline.

- Another limitation of our work is that it primarily focuses on open-domain question answering, which may could not be generalized to specialized domains.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (NSFC Grant No.62122089), Beijing Outstanding Young Scientist Program NO.BJJWZYJH012019100020098, and Intelligent Social Governance Platform, Major Innovation & Planning Interdisciplinary Platform for the "Double-First Class" Initiative, Renmin University of China, the Fundamental Research Funds for the Central Universities, the Research Funds of Renmin University of China, and the Ant Group Research Fund.

References

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1533–1544.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.

Yuhan Chen, Ang Lv, Ting-En Lin, Changyu Chen, Yuchuan Wu, Fei Huang, Yongbin Li, and Rui Yan. 2023. Fortify the shortest stave in attention: Enhancing context awareness of large language models for effective tool use.

Hao Cheng, Yelong Shen, Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2021. Unitedqa: A hybrid approach for open domain question answering. *arXiv preprint arXiv:2101.00178*.

W.-L Chiang, Z Li, and Z Lin. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *lmsys. org*.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*.

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

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*.

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*.

Omar Khattab, Christopher Potts, and Matei Zaharia. 2021. Relevance-guided supervision for openqa with colbert. *Transactions of the association for computational linguistics*, 9:929–944.

Nate Kushman, Yoav Artzi, Luke Zettlemoyer, and Regina Barzilay. 2014. Learning to automatically solve algebra word problems. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 271–281.

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.

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.

Shuang Liu, Dong Wang, Xiaoguang Li, Minghui Huang, and Meizhen Ding. 2022. A copy-augmented generative model for open-domain question answering. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 435–441.

Ang Lv, Jinpeng Li, Yuhan Chen, Gao Xing, Ji Zhang, and Rui Yan. 2023a. DialoGPS: Dialogue path sampling in continuous semantic space for data augmentation in multi-turn conversations. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1267–1280, Toronto, Canada. Association for Computational Linguistics.

Ang Lv, Jinpeng Li, Shufang Xie, and Rui Yan. 2023b. Envisioning future from the past: Hierarchical duality learning for multi-turn dialogue generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7382–7394, Toronto, Canada. Association for Computational Linguistics.

Ang Lv, Kaiyi Zhang, Shufang Xie, Quan Tu, Yuhan Chen, Ji-Rong Wen, and Rui Yan. 2023c. Are we falling in a middle-intelligence trap? an analysis and mitigation of the reversal curse.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2023. Self-refine: Iterative refinement with self-feedback. *arXiv preprint arXiv:2303.17651*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

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*.

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. 2019. Language models as knowledge bases? *arXiv preprint arXiv:1909.01066*.

Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. 2020. Rocketqa: An optimized training approach to dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2010.08191*.

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. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? *arXiv preprint arXiv:2002.08910*.

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.

Subhro Roy and Dan Roth. 2016. Solving general arithmetic word problems. *arXiv preprint arXiv:1608.01413*.

Noah Shinn, Beck Labash, and Ashwin Gopinath. 2023. Reflexion: an autonomous agent with dynamic memory and self-reflection. *arXiv preprint arXiv:2303.11366*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Yixuan Weng, Minjun Zhu, Shizhu He, Kang Liu, and Jun Zhao. 2022. Large language models are reasoners with self-verification. *arXiv preprint arXiv:2212.09561*.

Wenhao Yu, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, Soumya Sanyal, Chenguang Zhu, Michael Zeng, and Meng Jiang. 2022. Generate rather than retrieve: Large language models are strong context generators. *arXiv preprint arXiv:2209.10063*.

Kaiyi Zhang, Ang Lv, Yuhan Chen, Hansen Ha, Tao Xu, and Rui Yan. 2024. Batch-icl: Effective, efficient, and order-agnostic in-context learning.

Michael JQ Zhang and Eunsol Choi. 2021. Situatedqa: Incorporating extra-linguistic contexts into qa. *arXiv preprint arXiv:2109.06157*.

Fengbin Zhu, Wenqiang Lei, Chao Wang, Jianming Zheng, Soujanya Poria, and Tat-Seng Chua. 2021. Retrieving and reading: A comprehensive survey on open-domain question answering. *arXiv preprint arXiv:2101.00774*.

A Datasets and Baselines

A.1 Datasets

We conduct comprehensive experiments on three widely used benchmarks: NaturalQuestions (NQ, Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), WebQuestions (WebQ, Berant et al., 2013).

- **NQ**: comprises real queries that user issued on Google search engine along with answers.
- **TriviaQA**: consists of question-answer pairs collected from trivia and quiz-league websites
- **WebQ**: consists of questions selected using Google Suggest API, where the answers are entities in Freebase.

Statistics	NQ	TriviaQA	WebQ
Train	79168	78785	3478
Validation	8757	8837	300
Test	3610	11313	2032
Avg. Qlen	9.3	16.9	6.7
Avg. Alen	2.4	2.2	2.4

Table 3: Dataset splits and statistics.

A.2 Baselines

We compare our pipeline with the following strong baselines. (1) **BM25 + InstructGPT**: BM25 (Robertson et al., 1995) is a sparse retrieval method; (2) **Contriever + InstructGPT**: Contriever (Izacard et al., 2022) is an unsupervised dense retrieval model; (3) **Google + InstructGPT**; (4) **DPR + InstructGPT**: DPR (Karpukhin et al., 2020) is a supervised dense retrieval model and it trained on NQ, TriviaQA and WebQ datasets; (5) **InstructGPT (no docs.)** (Ouyang et al., 2022): InstructGPT is an LLM that usually serve as a reader or generator in ODQA; (6) **GENREAD** (Yu et al., 2022): GENREAD is the SoTA method in ODQA and is the first work that propose *generate-then-read* paradigm; (7) **Vanilla-United**: Moreover, in order to fully evaluate the effectiveness of our proposed method, we also compare our pipeline with another vanilla method which concatenates the two documents from retrieval and generation as contextual document directly.

All the baselines have the similar prompt template format for answer generation with a slight variation based on the number of supporting documents.

B Template Details

B.1 Template for Time-sensitivity

" Is the answer to the question depend on current time? Output with label: yes, no.\n\nQuestion: {*question*}\n\nThe label is "

B.2 Template for Document Selection

"You are a helpful and precise assistant for checking the quality of the statement.\n[Question]\n{*question*}\n\n[Statement 1]\n{*statement_1*}\n\n[Statement 2]\n{*statement_2*}\n\n[System]\n We would like to request your feedback on the quality of each statement to the user question displayed above.\n Please rate the factuality(according to wikipidia), relevance of each statement.\n\n Each statement receives an overall score on a scale of 1 to 10, where a higher score indicates better overall performance.\n Provide a comprehensive explanation of your evaluation, avoiding any potential bias and ensuring that the order in which the statement were presented does not affect your judgment. Output the better statement with '1', '2'. \n\n Output with the following format:\n The better statement is: *<1 or 2>*\n Evaluation evidence of statement: *<your evluation explanation here>*"

C Results

We provide a more detailed results in Table 4 including EM and F1 metric.

D Analysis

We provide a more detailed results in Table 5 including EM and F1 metric. And representative cases of three issues are in Table 6, Table 7, Table 8, respectively.

Models	NQ		TribiaQA		WebQ	
	F1	EM	F1	EM	F1	EM
*with retriever						
BM25+InstructGPT	-	19.7	-	52.2	-	15.8
Contriever+InstructGPT	-	18	-	51.3	-	16.6
Google+InstructGPT	-	28.8	-	58.8	-	20.4
DPR+InstructGPT*	39.1	<u>29.1</u>	65.1	55.7	34.8	21.5
*without retriever						
InstructGPT (no docs.)	-	20.9	-	57.5	-	18.6
GENREAD	40.6	28.2	68.9	59	<u>40.7</u>	<u>24.8</u>
*with retriever and generator						
Vanilla-United	<u>41.1</u>	28.1	<u>69.2</u>	<u>59.3</u>	37.1	20.9
FlexiQA	**44.5**	**32.4**	**70.0**	**60.5**	**40.8**	**25.1**

Table 4: EM and F1 score on NQ, TriviaQA and WebQ test sets.The best performance model is in **bold** and the second one is in <u>underline</u>.

Models	NQ				WebQ			
	TS set		non-TS set		TS set		non-TS set	
	EM	F1	EM	F1	EM	F1	EM	F1
DPR+InstructGPT	**22**	<u>30</u>	<u>30.3</u>	40.2	**14.1**	**22.2**	21.6	35.2
GENREAD	17.6	26.3	29.7	42.4	9.9	<u>21.7</u>	<u>25.2</u>	<u>41.4</u>
Vanilla-United	17	27.7	29.6	<u>42.8</u>	9.9	20.7	21.4	37.9
FlexiQA	<u>21.9</u>	**32.3**	**33.6**	**45.9**	<u>11.3</u>	20	**25.6**	**41.7**

Table 5: The experiment results of time-sensitive issue. TS means the time-sensitive subset of NQ and WebQ, while non-TS means the non-time-sensitive subset.

Question: Who is currently serving as president of the senate? **(NQ)**
Golden Answer: Orrin Hatch, Patrick Leahy
Generated document: The current president of the Senate is Kamala Harris, who was elected on January 3, 2021. Harris is the first woman to hold the office, as well as the first person of color.
Prediction: Kamala Harris.
Retrieved document: of another. This tradition has been observed without interruption since 1949. Since enactment of the current Presidential Succession Act in 1947, the president pro tempore is third in the line of succession to the presidency, after the vice president and the Speaker of the House of Representatives and ahead of the Secretary of State. The current President pro tempore of the Senate is Utah Republican Orrin Hatch. Elected on January 6, 2015, he is the 90th person to serve in this office...
Prediction: Orrin Hatch

Question: Who is the current presenter of BBC Radio 4 programme Desert Island Discs? **(TriviaQA)**
Golden Answer: Kirsty Young
Generated document: The current presenter of BBC Radio 4 programme Desert Island Discs is Lauren Laverne.
Prediction: Lauren Laverne.
Retrieved document: Desert Island Discs Desert Island Discs is a radio programme broadcast on BBC Radio 4. It was first broadcast on the BBC Forces Programme on 29 January 1942. Each week a guest, called a 'castaway' during the programme, is asked to choose eight recordings (usually, but not always, music), a book and a luxury item that they would take if they were to be cast away on a desert island, whilst discussing their lives and the reasons for their choices. It was devised and originally presented by Roy Plomley. Since 2006, the programme has been presented by Kirsty Young...
Prediction: Kirsty Young.

Question: Who is the minority leader of the house of representatives now? **(WebQ)**
Golden Answer: Nancy Pelosi
Generated document: The minority leader of the United States House of Representatives is the floor leader of the minority party. The current minority leader, Kevin McCarthy, is a Republican.
Prediction: Kevin McCarthy.
Retrieved document: the Speaker of the House. Typically, the Speaker does not participate in debate and rarely votes on the floor. In some cases, Majority Leaders have been more influential than the Speaker; notably, Tom DeLay was more prominent than Speaker Dennis Hastert. In addition, Speaker Newt Gingrich delegated to Dick Armey an unprecedented level of authority over scheduling legislation on the House floor. The current Minority Leader Nancy Pelosi serves as floor leader of the opposition party, and is the counterpart to the Majority Leader...
Prediction: Nancy Pelosi.

Table 6: Case studies of time-sensitive issue existing in *generate-then-read* paradigm. The three cases are all time-sensitive examples discriminated by LLM_{tsd}. The contents of documents that contradict to and relevant to the questions are marked red and blue respectively. We can see that our pipeline could handle time-sensitive issue effectively.

Question: Where was the original flight of the phoenix filmed? **(NQ)**
Golden Answer: Buttercup Valley, Arizona, 20th Century - Fox Studios, Pilot Knob Mesa, California
Generated document: The original Flight of the Phoenix was filmed in the Gobi Desert in Mongolia.
Prediction: the Gobi Desert in Mongolia.
Retrieved document: an oasis with a manned oil rig, the men celebrate and Towns and Dorfmann are reconciled. Principal photography started April 26, 1965, at the 20th Century-Fox Studios and 20th Century-Fox Ranch, California. Other filming locations, simulating the desert, were Buttercup Valley, Arizona and Pilot Knob Mesa, California. The flying sequences were all filmed at Pilot Knob Mesa near Winterhaven, located in California's Imperial Valley, on the western fringes of Yuma, Arizona. In 2005, Hollywood aviation historian Simon Beck identified the aircraft used in the film: The C-82As were from Steward-Davies Inc....
Prediction: Pilot Knob Mesa.

Question: Whose "left hand" rule can predict the direction of motion of an electric motor? **(TriviaQA)**
Golden Answer: Fleming, Sir John Ambrose, Sir John Ambrose Fleming, Sir Ambrose Fleming, Ambrose Fleming, J. A. Fleming, John Ambrose Fleming
Generated document: The rule is named after its discoverer, Michael Faraday. It can be stated as follows: "If a conductor is moved towards a magnet, an electric current will flow in it; if the conductor is moved away from the magnet, the current will stop."
Prediction: Michael Faraday's "left hand" rule.
Retrieved document: Fleming's left-hand rule for motors Fleming's left-hand rule for electric motors is one of a pair of visual mnemonics, the other being Fleming's right-hand rule (for generators).They were originated by John Ambrose Fleming, in the late 19th century, as a simple way of working out the direction of motion in an electric motor motor, or the direction of electric current in an electric generator...
Prediction: John Ambrose Fleming's.

Question: Who plays riley on buffy the vampire slayer? **(WebQ)**
Golden Answer: Marc Blucas
Generated document: Sarah Michelle Gellar plays the title character, Buffy Summers, on the television series Buffy the Vampire Slayer (1997-2003). Summers is a Slayer, one in a long line of young women chosen by fate to battle evil forces.
Prediction: unknown.
Retrieved document: Riley Finn Riley Finn is a fictional character created by Joss Whedon for the television series "Buffy the Vampire Slayer". Portrayed by Marc Blucas, Riley was introduced in the 1999 season four premiere episode, "The Freshman", and Blucas was part of the series credited cast for the second part of season four and the first part of season five. Most notably, Riley is one of three long-term romantic interests for series' heroine Buffy Summers (Sarah Michelle Gellar)...
Prediction: Marc Blucas

Table 7: Case studies of factual issue existing in *generate-then-read* paradigm. The contents of documents that contradict to and relevant to the questions are marked red and blue respectively. The three cases show that our pipeline choose the retrieval documents as evidences when generation method give the non-factual evidences, which means that our pipeline could handle factual issue effectively.

Question: In which sea pearl is found in india? **(NQ)**
Golden Answer: the Indian Ocean
Generated document: The sea pearl is found in the Indian Ocean.It is a type of pearl that is found in the oysters that live in the ocean. The sea pearl is a very valuable type of pearl and is used in many different types of jewelry.
Prediction: the Indian Ocean.
Retrieved document: the Indian Ocean. Chinese possessions in the Indian Ocean consist primarily of commercial ports owned and operated by Chinese firms, as well as resupply stations operating in agreement with the Chinese central government. The two largest projects consist of a Chinese-financed commercial shipping center in Hambantota, Sri Lanka, and a Chinese-controlled deep-water port near the mouth of the Persian Gulf in Gwadar, Pakistan, which is also the crux of China's massive $46 billion China Pakistan Economic Corridor project...
Prediction: not found in the passage.

Question: If you were performing an orison what would you be doing? **(TriviaQA)**
Golden Answer: Prayed, Prayers, Pray, Prayer group, Orison (spirituality), Orisons, Prayer, Namāz, Daily prayer, Daily prayers, Devotional prayer, Praying, Prayer in Buddhism
Generated document: Orison is a type of prayer.
Prediction: praying.
Retrieved document: first three lines which include the words 'investigate', 'insinuate', 'intimidate', 'complicate', 'wait', 'hesitate', 'state', 'fate' and 'awaits'. The song fades out with lead singer Phil Lynott doing an impersonation of Elvis Presley which seemed to have little to do with the meaning of the song. Guitarist Scott Gorham suggested, "I think he just wanted to use his American accent there." The single reached No. 14 in the British charts and remained in the charts for nine weeks. It reached No. 25 in Ireland, charting for two weeks. An accompanying video was filmed at Molineir Studios, directed by David Mallet...
Prediction: not provided in the passage.

Question: Who was vice president after kennedy died? **(WebQ)**
Golden Answer: Lyndon B. Johnson
Generated document: Lyndon B. Johnson was the 36th President of the United States, serving from 1963 to 1969. He had served as the 37th Vice President of the United States from 1961 to 1963 and assumed the presidency following the assassination of President John F. Kennedy. A Democrat from Texas, Johnson also served as a United States Representative and as the Majority Leader in the United States Senate. During his presidency, Johnson was responsible for designing the "Great Society" legislation that included laws that created Medicare, Medicaid, environmental protection, civil rights, urban renewal, aid to education, and public broadcasting. Johnson was famed for his domineering personality and the "Johnson treatment", his aggressive coercion of powerful politicians to advance legislation.
Prediction: Lyndon B. Johnson.
Retrieved document: United States vice presidential selection, 1974 In 1974, Republican President Richard Nixon was forced to resign following the Watergate Scandal. Vice President Gerald Ford ascended to the presidency, leaving the office of vice president vacant. Under the terms of the 25th Amendment, a vice presidential vacancy is filled when the president nominates a candidate who is confirmed by both houses of Congress. On August 20, 1974, Ford announced his nomination of former New York Governor Nelson Rockefeller to fill the vacancy. Ford also considered picking Tennessee Senator Howard Baker and former Republican National Committee Chairman George H.W. Bush...
Prediction: Gerald Ford.

Table 8: Case studies of irrelevance issue existing in *retrieve-then-read* paradigm. The contents of documents that contradict to and relevant to the questions are marked red and blue respectively. The three cases show that our pipeline choose the generated documents as evidences when retrieved documents have no relation with questions, which means that our pipeline could handle irrelevance issue effectively.