Arxiv Copilot: A Self-Evolving and Efficient LLM System for Personalized Academic Assistance

Guanyu Lin[1 2*], Tao Feng[1*], Pengrui Han[1 3*], Ge Liu[1], Jiaxuan You[1]
[1]University of Illinois at Urbana-Champaign, [2]Carnegie Mellon University, [3]Carleton College
[*]Equal Contribution

Abstract

As scientific research proliferates, researchers face the daunting task of navigating and reading vast amounts of literature. Existing solutions, such as document QA, fail to provide personalized and up-to-date information efficiently. We present Arxiv Copilot, a self-evolving, efficient LLM system designed to assist researchers, based on thought-retrieval, user profile and high performance optimization. Specifically, Arxiv Copilot can offer personalized research services, maintaining a real-time updated database. Quantitative evaluation demonstrates that Arxiv Copilot saves 69.92% of time after efficient deployment. This paper details the design and implementation of Arxiv Copilot, highlighting its contributions to personalized academic support and its potential to streamline the research process. We have deployed Arxiv Copilot at: https://huggingface.co/spaces/ulab-ai/ArxivCopilot.

1 Introduction

As scientific research has proliferated at an unprecedented rate, researchers are now supposed to navigate and interpret vast amounts of published and pre-print papers (Tenopir et al., 2009). Indeed, researchers need to keep up with the latest trend. This involves continuously searching for relevant papers, quickly evaluating which papers for thorough reading, analyzing trending research topics, and reflecting potential ideas. Therefore, they should dedicate significant time to following up the latest papers. However, the large volume of papers make it hard for them to locate the related information, resulting in the waste of time.

Fortunately, based on retrieval-augmented generation (RAG) (Weijia et al., 2023), LLMs (Zhao et al., 2023) can help to extract and summarize useful information from such external papers (Chen et al., 2023). Thus, the above background leads us to a crucial question: *How can we design a LLM system that can assist researchers in obtaining the latest research information from massive papers?*

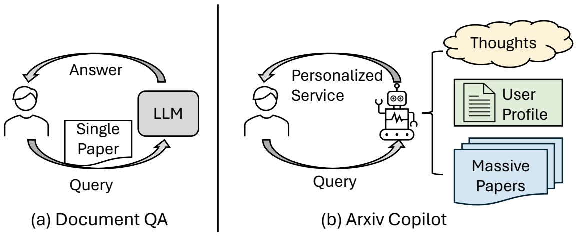


Figure 1: **Comparison of (a) document Question Answering (QA) with our (b) Arxiv Copilot.** Conventional document QA tends to help user understand the content of specific paper while our Arxiv Copilot can further act like a real research assistant who can provide personalized service based on user profile.

To provide intelligent assistance for researchers, existing works have targeted several tasks, such as skimming (Fok et al., 2023), searching (Ammar et al., 2018; Beel and Gipp, 2009), and reading (Head et al., 2021). However, these approaches focus either on understanding the content of paper document (as shown in Figure 1 (a)) or improving the ranking of relevant papers. They fall short of acting like a real researcher who can get *personalized* and *up-to-date* information on demand. Moreover, as researchers read more papers, they become increasingly experienced—a characteristic that current systems fail to replicate through *self-evolution*. Finally, *efficiency* remains a critical challenge in retrieving and extracting useful information from the vast and continuously growing pool of papers.

To address the above challenges, we develop Arxiv Copilot, a self-evolving and efficient LLM system for personalized academic assistance. More specifically, Arxiv Copilot can provide personalized research service, self-evolve like a human researcher as shown in Figure 1 (b), and make prompt responses. The detailed characteristics of Arxiv Copilot are as below.

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing
System Demonstrations, pages 122–130
November 12-16, 2024 ©2024 Association for Computational Linguistics

- **Personalized research service**. Arxiv Copilot can provide personalized research assistance based on user profile. Specifically, it can (1) derive your profile from your historical publications, (2) analyze the latest trending research topics and provide ideas (which will be sent with email if sign up), and (3) offer research chat and advisory services.

- **Real-time updated research database**. Arxiv Copilot could refresh its paper database daily from the latest Arxiv papers. Users further have the option to select a date range to query the papers.

- **Self-evolved thought retrieval**. Arxiv Copilot enhances the response of LLM based on a thought retrieval (Feng et al., 2024) method, which will self-evolve based on the historical user query.

- **High performance optimization**. Arxiv Copilot employs a real-time feature pool for efficient retrieval, a multithreading engine for effective memory management and I/O, and a cache to store responses with a high probability of re-querying. These optimizations significantly reduce API cost and response time by 69.92%.

More importantly, user comment feedback indicates that Arxiv Copilot can save researchers at least 20 minutes in obtaining the same amount of information. This demonstrates that Arxiv Copilot not only provides valuable academic assistance but also saves researchers' time. Our evaluations, both quantitative and qualitative, further highlight its superiority in efficiency and user experience. Specifically, we reduce 69.92% of time cost after efficient deployment. In summary, this work presents the following *contributions*:

- We design Arxiv Copilot, a self-evolving demo that provides personalized academic services based on real-time updated Arxiv papers.

- We improve the efficiency and scalability of Arxiv Copilot through retrieval feature pre-computation, parallel computation, asynchronous I/O, and frequent query caching.

- We evaluate the proposed Arxiv Copilot from both qualitative and quantitative perspectives.

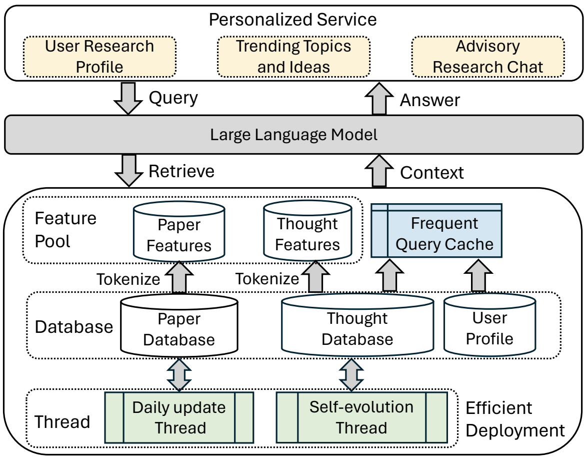


Figure 2: **Architecture of Arxiv Copilot from bottom-to-up perspective.** (a) In personalized service, Arxiv Copilot provides interactive services including the generation of user research profile, analysis of research trends and ideas, and advisory chatting about research. (b) In large language model, user demand from interaction will be used for retrieving and collecting relevant context, and then LLM will generate answer and make response to user demand. (c) In efficient deployment, feature pre-computation, parallel computation and caching techniques are applied to speed up the retrieval process and guarantee the efficient response.

2 Arxiv Copilot

As shown in Figure 2, our proposed Arxiv Copilot mainly consists of the following four key parts:

- **Personalized Service**. This part aims to generate personalized response based on user demand, including the generation of user research profile, analysis of personalized trending research topics or ideas with email, and personalized chat about research advisory.

- **Real-time Updating**. This part allows for the daily updating of its database using the latest Arxiv papers. Additionally, users can specify a range of time for papers to be retrieved.

- **Self-evolution**. This part improves LLM responses using a thought retrieval technique that adapts and evolves from past user queries.

- **Efficient Deployment**. This part achieves efficient deployment by a constantly updating feature pre-computation node for swift retrieval, a high performance engine for memory and I/O management, and a cache for storing frequently queried responses.

For the detailed description of them, we will introduce in the subsequent section.

2.1 Personalized Service

User Research Profile In user research profile, each user $u \in \mathcal{U}$ can input his/her name n_u to get historical publication as: $\mathcal{D}_{u,:t-1} \leftarrow \textbf{Search}\left(n_u\right)$. Here **Search**() is the search method based on Arxiv API (). The retrieved papers $\mathcal{D}_{u,:t-1}$ will then be fed into LLM for profile generation as below.

$$\mathcal{P}_{u,t} \leftarrow \textbf{LLM}\left(\text{Instruct}_p, \mathcal{D}_{u,:t-1}\right). \tag{1}$$

where $\mathcal{P}_{u,t}$ is the generated profile for user u at time step t. Besides, Instruct_p is the instruction for profile generation, which is defined in Section 1.

Trending Topics and Ideas To further get the personalized trending research topics based on user profile, we firstly can retrieve some papers related to user profile $\mathcal{P}_{u,t}$, as follows:

$$\mathcal{R}^{trend}_{u,t} \leftarrow \textbf{Rtri}\left(\textbf{Tkn}\left(\mathcal{P}_{u,t}\right), \textbf{Tkn}\left(\mathcal{D}_{:,:t-1}\right)\right), \tag{2}$$

where $\mathcal{R}^{trend}_{u,t}$ are the retrieved papers related to user profile. Besides, **Rtri**() and **Tkn**() are the methods for retrieval and tokenization. Based on the retrieved papers $\mathcal{R}^{trend}_{u,t}$, we can then feed them into LLM to generate the personalized trending research topics as below.

$$\mathcal{C}_{u,t} \leftarrow \textbf{LLM}\left(\text{Instruct}_t, \mathcal{R}^{trend}_{u,t}\right) \tag{3}$$

where $\mathcal{C}_{u,t}$ are the personalized trending research topics and Instruct_t is the instruction for research topic generation defined at Section 2. With the personalized trending research topics, we can finally get some ideas related to the research topics of user u, as:

$$\mathcal{I}_{u,t} \leftarrow \textbf{LLM}\left(\text{Instruct}_i, \mathcal{C}_{u,t}\right), \tag{4}$$

where $\mathcal{I}_{u,t}$ are the research ideas related to the personalized trending research topics $\mathcal{C}_{u,t}$ of user u. Here Instruct_i is the instruction for idea generation defined at Section 3. Besides, we also provide weekly report service for trending topics and ideas if users sign up with email.

Advisory Research Chat In advisory research chat, user can further input his/her question $\mathcal{Q}_{u,t}$ and get personalized assistance based on previous generated trends and ideas. Firstly, we need to retrieve historical papers and generated contents $\mathcal{R}^{chat}_{u,t}$ related to the input question as:

$$\mathcal{R}^{chat}_{u,t} \leftarrow \textbf{Rtri}(\textbf{Tkn}\left(\mathcal{Q}_{u,t}\right), [\textbf{Tkn}\left(\mathcal{D}_{:,:t-1}\right), \textbf{Tkn}\left(\mathcal{B}_{:,:t-1}\right)]), \tag{5}$$

where $\mathcal{B}_{:,:t-1} = \mathcal{C}_{:,:t-1} \cup \mathcal{I}_{:,:t-1} \cup \mathcal{A}_{:,:t-1}$ is the thought database including generated research trends $\mathcal{C}_{:,:t-1}$, ideas $\mathcal{I}_{:,:t-1}$, and answers $\mathcal{A}_{:,:t-1}$. Based on the retrieved historical papers and generated contents, we can then feed them into LLM for answering:

$$\mathcal{A}_{u,t} \leftarrow \textbf{LLM}\left(\mathcal{Q}_{u,t}, \mathcal{R}^{chat}_{u,t}, \mathcal{P}_{u,t}\right) \tag{6}$$

where $\mathcal{A}_{u,t}$ is the answer for user u based on his/her question $\mathcal{Q}_{u,t}$. Here feeding $\mathcal{P}_{u,t}$ into LLM means the generated answer will be organized in a personalized manner related to the profile of user u.

2.2 Real-time Updating

Daily Updating During daily updating, Arxiv Copilot will download the newest papers from Arxiv and refresh the paper storage as: $\mathcal{D}_{:,:t} \leftarrow \mathcal{D}_{:,:t-1} \cup \mathcal{D}_{:,t}$, where $\mathcal{D}_{:,t}$ are the newest papers and $\mathcal{D}_{:,:t}$ is the refreshed paper storage.

Time Range Selection As users may not care about some old papers and trends. Thus, in time range selection, users can select the daily papers $\mathcal{D}_{:,t}$, weekly papers $\mathcal{D}_{:,t-6:t}$, and all papers $\mathcal{D}_{:,:t}$ for personalized research trend and idea generation.

2.3 Self-evolution

As human researchers will become more and more experienced, Arxiv Copilot also evolves its thought by incorporating the interacted contents with users as below.

$$\begin{aligned} \mathcal{A}_{:,:t} &\leftarrow \mathcal{A}_{:,:t-1} \cup \mathcal{A}_{:,t}, \\ \mathcal{C}_{:,:t} &\leftarrow \mathcal{C}_{:,:t-1} \cup \mathcal{C}_{:,t}, \\ \mathcal{I}_{:,:t} &\leftarrow \mathcal{I}_{:,:t-1} \cup \mathcal{I}_{:,t}, \end{aligned} \tag{7}$$

where $\mathcal{A}_{:,:t}$, $\mathcal{C}_{:,:t}$, and $\mathcal{I}_{:,:t}$ are the self-evolved thought at time step t by incorporating answers, research trends and ideas interacted with users. That is to say, the more interactions with users, the smarter Arxiv Copilot will be.

2.4 Efficient Deployment

Feature Pre-computation In feature pre-computation, we construct a feature pool and pre-compute the paper embedding $\mathbf{D}_{:,:t-1}$ and thought embedding $\mathbf{B}_{:,:t-1}$ for retrieval. By this way, we do not need to re-tokenize the input text while retrieval, which saves a lot of time. Thus the retrieval equations at Eq. (2) and (5), respectively, can be reformulated as Eq. (8) and (9).

$$\mathcal{R}^{trend}_{u,t} \leftarrow \textbf{Rtri}\left(\textbf{Tkn}\left(\mathcal{P}_{u,t}\right), \mathbf{D}_{:,:t-1}\right), \tag{8}$$

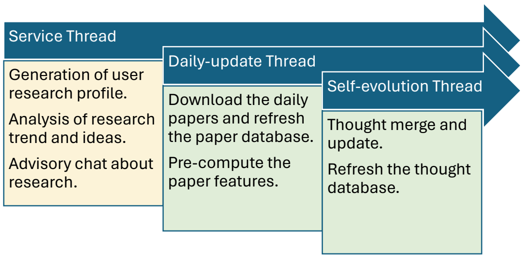

Service Thread
Generation of user research profile.
Analysis of research trend and ideas.
Advisory chat about research.
Daily-update Thread
Download the daily papers and refresh the paper database.
Pre-compute the paper features.
Self-evolution Thread
Thought merge and update.
Refresh the thought database.

Figure 3: **Multi-thread engine keeps Arxiv Copilot service away from waiting for daily updating of papers and self-evolution of thoughts.** The daily-update thread and self-evolution thread will achieve thought memory management and asynchronous I/O without disturbing the service thread.

$$\mathcal{R}_{u,t}^{chat} \leftarrow \mathbf{Rtri}\left(\mathbf{Tkn}\left(\mathcal{Q}_{u,t}\right),\left[\mathbf{D}_{:,:t-1},\mathbf{B}_{:,:t-1}\right]\right), \quad (9)$$

where the computational costs for the tokenization methods on papers $\mathcal{D}_{:,:t-1}$ and thought $\mathcal{B}_{:,:t-1}$ are saved. Besides, the paper embedding and thought embedding will be updated through:

$$\mathbf{D}_{:,:t} \leftarrow [\mathbf{D}_{:,:t-1}, \mathbf{Tkn}\left(\mathcal{D}_{:,t}\right)], \quad (10)$$

$$\begin{aligned}\mathbf{A}_{:,:t} &\leftarrow [\mathbf{A}_{:,:t-1}, \mathbf{Tkn}\left(\mathcal{A}_{:,t}\right)],\\ \mathbf{C}_{u,:t} &\leftarrow [\mathbf{C}_{u,:t-1}, \mathbf{Tkn}\left(\mathcal{C}_{:,t}\right)],\\ \mathbf{I}_{u,:t} &\leftarrow [\mathbf{I}_{u,:t-1}, \mathbf{Tkn}\left(\mathcal{I}_{:,t}\right)],\\ \mathbf{B}_{:,:t} &\leftarrow [\mathbf{A}_{:,:t}, \mathbf{C}_{:,:t}, \mathbf{I}_{:,:t}],\end{aligned} \quad (11)$$

where $\mathbf{D}_{:,:t}$ and $\mathbf{B}_{:,:t}$ are the updated paper embedding and thought embedding, respectively.

Multi-threading Engine As our Arxiv Copilot needs to refresh the database and update thoughts frequently, the user interactive service will be disturbed and become inefficient. Thus we further implement a multi-thread engine as Figure 3 to reduce the waiting time of interactive service when updating. Specifically, it consists of service thread, daily-update thread and self-evolution thread to execute the personalized service, paper updating and thought management at the same time. With such multi-thread engine, there is no need for the main personalized service to wait for storage refreshing. That is to say, all memory management processes and I/O processes will be finished in parallel.

Frequent Query Cache In frequent query cache, we store the content that will be frequently queried at hash cache. More specifically, user profile, research trends and ideas may will stay unchanged within a period of time. Thus these static contents are more likely to be re-queried, and we store them in hash cache $\mathbf{Hash}()$ as:

$$\begin{aligned}&\mathcal{P}_{u,t} \leftarrow \mathbf{Hash}\left(n_u\right), \mathcal{C}_{u,t} \leftarrow \mathbf{Hash}\left(\mathcal{P}_{u,t}\right),\\ &\mathcal{I}_{u,t} \leftarrow \mathbf{Hash}\left(\mathcal{P}_{u,t}\right), \mathcal{R}_{u,t}^{trend} \leftarrow \mathbf{Hash}\left(\mathcal{P}_{u,t}\right),\end{aligned} \quad (12)$$

where $\mathcal{R}_{u,t}^{trend}$ are the papers we retrieve for research trend generation. As $\mathcal{R}_{u,t}^{trend}$ will also be presented at Arxiv Copilot as trending papers, we hash them in the cache. With this hash cache, we can make instant responses when contents are re-queried.

3 User Guidance and Usage

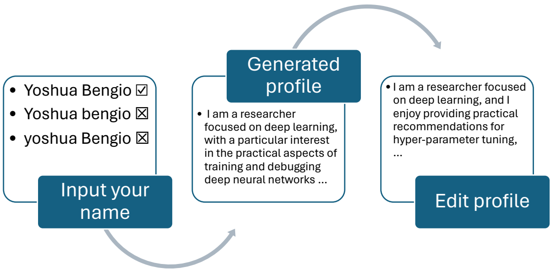
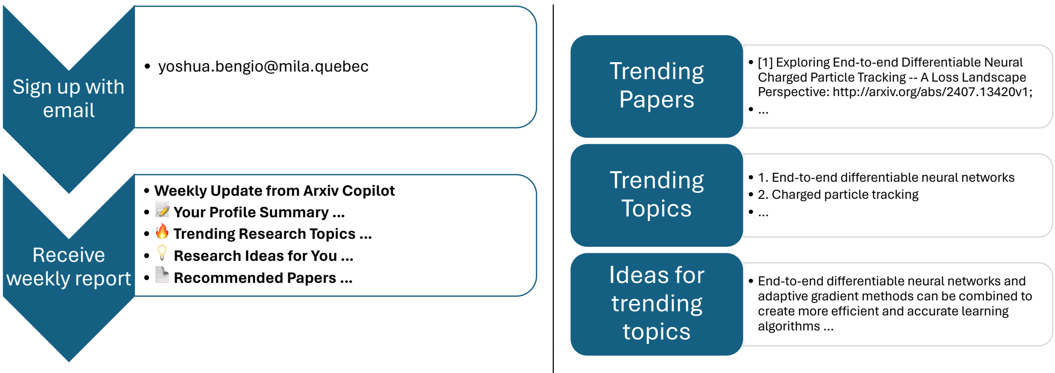

Yoshua Bengio ☑
Yoshua bengio ☒
yoshua Bengio ☒
Input your name
Generated profile
I am a researcher focused on deep learning, with a particular interest in the practical aspects of training and debugging deep neural networks ...
I am a researcher focused on deep learning, and I enjoy providing practical recommendations for hyper-parameter tuning, ...
Edit profile

Figure 4: **Flowchart for the interaction of user research profile in Arxiv Copilot**. Users can input his/her name to generate the personalized profile based on historical publication. Besides, if users are unsatisfied with the generated profile or fail to get historical publication, they also can manually edit the profile.

User Research Profile In "Set your profile!", as shown in Figure 4, we have input text box "Input your name:" where user can input his/her name and then click button "Set Profile" to obtain the profile from output text box "Generated profile (can be edited):". Here the output text box of generated profile also can be modified and edited by clicking button "Edit Profile". The details of each button operation is shown in Figure 9 of Appendix A.

Trending Topics and Ideas In "Get trending topics and ideas!", as shown in Figure 5, user can sigu up to get the weekly update of trending research topics, ideas and papers. Besides, user can also select the time range and then click button "Confirm" to filter out papers from daily, weekly and all historical publication time. Then in the "Trending Papers", "Trending Topics" and "Ideas for Trending Topic" text boxes, respectively, personalized trending papers, topics and ideas related to the user will


Sign up with email
- yoshua.bengio@mila.quebec
Receive weekly report
- Weekly Update from Arxiv Copilot
- Your Profile Summary ...
- Trending Research Topics ...
- Research Ideas for You ...
- Recommended Papers ...
Trending Papers
- [1] Exploring End-to-end Differentiable Neural Charged Particle Tracking -- A Loss Landscape Perspective: http://arxiv.org/abs/2407.13420v1;
- ...
Trending Topics
- 1. End-to-end differentiable neural networks
- 2. Charged particle tracking
- ...
Ideas for trending topics
- End-to-end differentiable neural networks and adaptive gradient methods can be combined to create more efficient and accurate learning algorithms ...

(a) Sign up with email

(b) Get research trend

Figure 5: **Diagram for the interaction of research trend and ideas in Arxiv Copilot**. (a) Users can sign up with email to receive the weekly update. (b) Besides, users can also select the time range for getting the daily, weekly or all historical research trend.

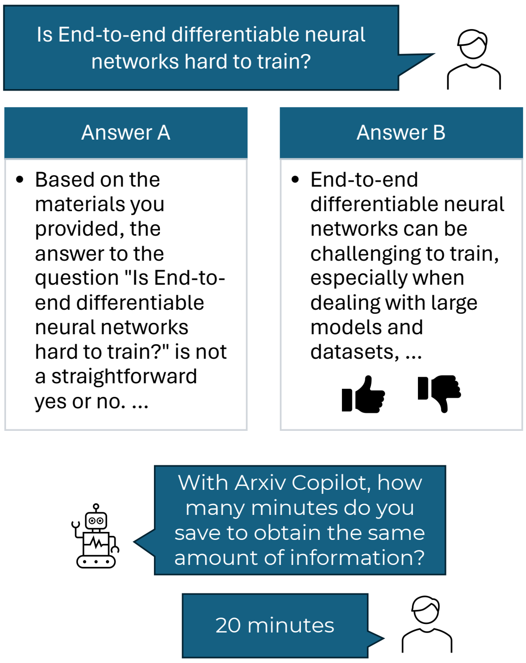

Is End-to-end differentiable neural networks hard to train?
Answer A
- Based on the materials you provided, the answer to the question "Is End-to-end differentiable neural networks hard to train?" is not a straightforward yes or no. ...
Answer B
- End-to-end differentiable neural networks can be challenging to train, especially when dealing with large models and datasets, ...
With Arxiv Copilot, how many minutes do you save to obtain the same amount of information?
20 minutes

Figure 6: **Diagram for the interaction of advisory research chat in Arxiv Copilot**. After users ask the question, Arxiv Copilot will give two answers. Specifically, the first answer is with both thought and paper retrieval while the second answer is just with paper retrieval. Here the second answer will have two feedback choices for users, one is 'like' and another is 'dislike'. If users click 'like', the first answer will be removed. Otherwise, the second answer will removed. Besides, users can also provide feedback on the saved time.

be presented. The details of each button operation is shown in Figure 10 of Appendix A.

Advisory Research Chat In "Chat with Arxiv Copilot!", as shown in Figure 6, user can chat with arxiv copilot by typing the question into the input text box of Chatbot and then click button "Send" or enter "carriage return" in the keyboard. Then Arxiv Copilot will return with two candidate answers, the first answer is based on thought and paper retrieval while the second answer is just based on paper retrieval. Here user can give feedback and choose the preferred answer with either augmented thoughts or just initial papers. Besides, by clicking the button "Clear", user can clean all historical chat with Arxiv Copilot. Finally, user can give further feedback about how many minutes Arxiv Copilot has helped you to save time in research by clicking button "Comment". The details of each button operation is shown in Figure 11 of Appendix A.

4 Evaluation

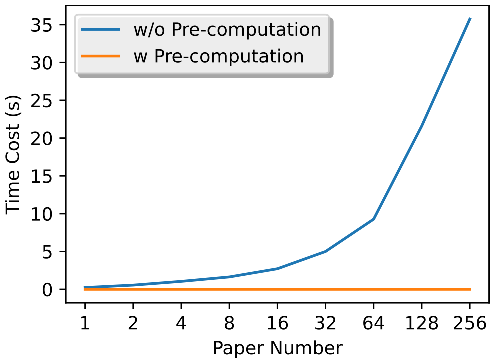


Figure 7: **Feature pre-computation significantly improves the efficiency.** The time cost for retrieval without feature pre-computation will grow with the exponential increase of paper number, while our proposed feature pre-computation stays unchanged and keeps constant time cost.

Quantitative: Efficiency Firstly, as shown in Figure 7, we plot the time costs of paper retrieval without feature pre-computation and with pre-computation. From the result, we can discover that our proposed feature pre-computation is very efficient, which has a constant computational cost at $O(1)$. However, the time cost of retrieval without pre-computation will grow significantly with the increase of papers. This is because there is no need to re-tokenization on contents to be retrieved under feature pre-computation, while those without pre-computation will repeatedly tokenize the contents each time.

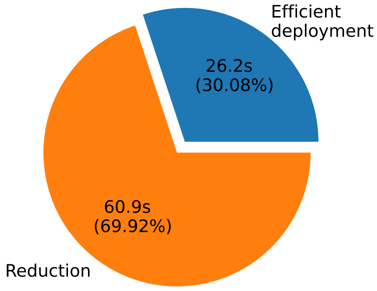


Figure 8: **Efficient deployment methods dramatically reduce the time cost.** The average total time cost before efficient deployment is 87.1s (26.2s + 60.9s), which is reduced by 69.92% after efficient deployment.

Besides, we also plot the pie chart of time cost reduced by efficient deployment and that under efficient deployment as Figure 8. Specifically, we can see that our efficient deployment reduces the total time cost average by 60.9s. And now is just requires average 26.2s for making response, which improves the user experience a lot compared with initial 87.1s.

Qualitative: User Study After collecting the user feedback from advisory research chat, we find that there are about 75% of users will prefer the answers with self-evolution augmentation, illustrating the effectiveness of Arxiv Copilot for self-evolving like real human researchers.

However, there is still a small problem. That is, when user inputs his/her name in profile generation, there may be duplicate. For example, when you input "Feifei Li", you will get the profile of a researcher in quantum computing, instead of the researcher in artificial intelligence. In such case, the users may need to input and edit the profile manually by themselves.

5 Related Work

Retrieval Augmented Generation Retrieval Augmented Generation (RAG) (Lewis et al., 2020) augments LLMs by retrieving and incorporating external context and information. Existing approaches employ methods can be classified into the following categories: embedding-based method (Izacard et al., 2022; Lin et al., 2023), fine-tuning re-ranker method (Ram et al., 2023) and keyword-based method (Robertson et al., 2009). While these strategies have shown decent outcomes, they still face many challenges in the extremely long context. Fortunately, hierarchical tree-based method (Chen et al., 2023) and thought-retrieval method (Feng et al., 2024) can well address these challenges. Though extending the long context window, existing method is still inefficient when encoding the extremely long context. Thus, in this work, we further improve the efficiency of long-context RAG by feature pre-computation and several high performance computing techniques.

Academic Assistance with Language Models Language models can provide academic assistance based on scientific papers in variety of ways. Firstly, it can make summary of the paper's content to help understanding (Nenkova and McKeown, 2012; Sefid and Giles, 2022). Besides, it also can help researchers to skim today's emerging papers (Fok et al., 2023) and read useful information (August et al., 2023). However, existing works mainly focus on single paper understanding. Unlike them, Arxiv Copilot further provides personalized academic assistance like a human researcher.

6 Conclusion and Future Work

To address the challenges posed by the rapid growth of scientific research, we propose Arxiv Copilot with a personalized, self-evolving, and efficient LLM system. It offers tailored research services, maintains a real-time updated database, and employs advanced optimization techniques to enhance performance. Evaluations demonstrate its ability to significantly reduce the time researchers spend on literature review while improving accuracy and user experience. By setting a new standard for personalized academic support, Arxiv Copilot stands as a valuable tool for the scientific community, enhancing the research process. Future work will focus on integrating additional sources beyond Arxiv to provide a broader research perspective.

References

Waleed Ammar, Dirk Groeneveld, Chandra Bhagavatula, Iz Beltagy, Miles Crawford, Doug Downey, Jason Dunkelberger, Ahmed Elgohary, Sergey Feldman, Vu Ha, et al. 2018. Construction of the literature graph in semantic scholar. *arXiv preprint arXiv:1805.02262*.

Tal August, Lucy Lu Wang, Jonathan Bragg, Marti A Hearst, Andrew Head, and Kyle Lo. 2023. Paper plain: Making medical research papers approachable to healthcare consumers with natural language processing. *ACM Transactions on Computer-Human Interaction*, 30(5):1–38.

Jöran Beel and Bela Gipp. 2009. Google scholar's ranking algorithm: an introductory overview. In *Proceedings of the 12th international conference on scientometrics and informetrics (ISSI'09)*, volume 1, pages 230–241. Rio de Janeiro (Brazil).

Howard Chen, Ramakanth Pasunuru, Jason Weston, and Asli Celikyilmaz. 2023. Walking down the memory maze: Beyond context limit through interactive reading. *arXiv preprint arXiv:2310.05029*.

Tao Feng, Pengrui Han, Guanyu Lin, Ge Liu, and Jiaxuan You. 2024. Thought-retriever: Don't just retrieve raw data, retrieve thoughts. In *ICLR 2024 Workshop: How Far Are We From AGI*.

Raymond Fok, Hita Kambhamettu, Luca Soldaini, Jonathan Bragg, Kyle Lo, Marti Hearst, Andrew Head, and Daniel S Weld. 2023. Scim: Intelligent skimming support for scientific papers. In *Proceedings of the 28th International Conference on Intelligent User Interfaces*, pages 476–490.

Andrew Head, Kyle Lo, Dongyeop Kang, Raymond Fok, Sam Skjonsberg, Daniel S Weld, and Marti A Hearst. 2021. Augmenting scientific papers with just-in-time, position-sensitive definitions of terms and symbols. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–18.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning. *Preprint*, arXiv:2112.09118.

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.

Sheng-Chieh Lin, Akari Asai, Minghan Li, Barlas Oguz, Jimmy Lin, Yashar Mehdad, Wen-tau Yih, and Xilun Chen. 2023. How to train your dragon: Diverse augmentation towards generalizable dense retrieval. *arXiv preprint arXiv:2302.07452*.

Ani Nenkova and Kathleen McKeown. 2012. A survey of text summarization techniques. *Mining text data*, pages 43–76.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *arXiv preprint arXiv:2302.00083*.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Athar Sefid and C Lee Giles. 2022. Scibertsum: extractive summarization for scientific documents. In *International workshop on document analysis systems*, pages 688–701. Springer.

Carol Tenopir, Donald W King, Sheri Edwards, and Lei Wu. 2009. Electronic journals and changes in scholarly article seeking and reading patterns. In *Aslib proceedings*, volume 61, pages 5–32. Emerald Group Publishing Limited.

Shi Weijia, Min Sewon, Yasunaga Michihiro, Seo Minjoon, James Rich, Lewis Mike, and Yih Wen-tau. 2023. Replug: Retrieval-augmented black-box language models. *ArXiv: 2301.12652*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

A Example Appendix

Table 1: Prompts for profile generation.

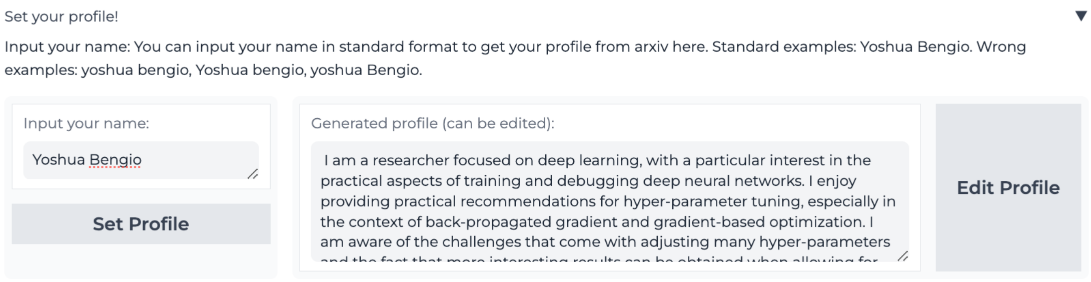
Instruction: Based on the list of the researcher's papers from different periods, please write a comprehensive first person persona. Focus more on recent papers. Be concise and clear (around 300 words). Here are the papers from different periods: {papers}

Table 2: Prompts for trending research topic generation.

Instruction: Given some recent paper titles and abstracts. Could you summarize no more than 10 top keywords of high level research backgrounds and trends. Here are the retrieved paper abstracts: {papers}

Table 3: Prompts for research idea generation.

Instruction: Here is a high-level summarized trend of a research field: {trend} How do you view this field? Do you have any novel ideas or insights? Please give me 3 to 5 novel ideas and insights in bullet points. Each bullet points should be concise, containing 2 or 3 sentences.



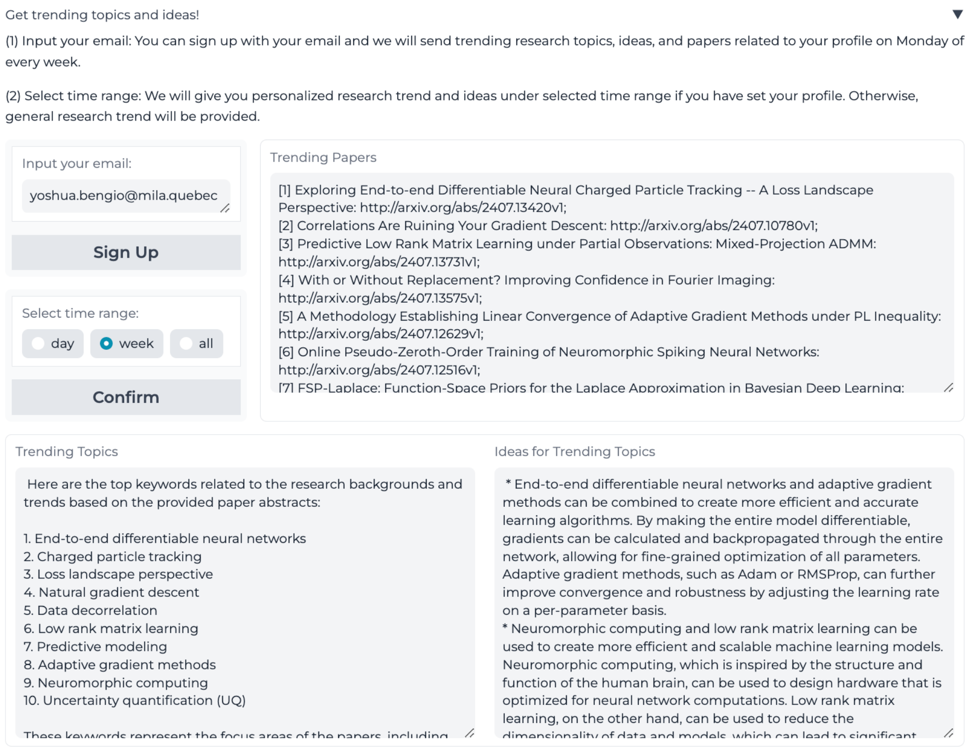
Set your profile!

Input your name: You can input your name in standard format to get your profile from arxiv here. Standard examples: Yoshua Bengio. Wrong examples: yoshua bengio, Yoshua bengio, yoshua Bengio.

Input your name: Yoshua Bengio

Generated profile (can be edited): I am a researcher focused on deep learning, with a particular interest in the practical aspects of training and debugging deep neural networks. I enjoy providing practical recommendations for hyper-parameter tuning, especially in the context of back-propagated gradient and gradient-based optimization. I am aware of the challenges that come with adjusting many hyper-parameters and the fact that more interesting results can be obtained when allowing for

Figure 9: **Screenshot for the interaction of user research profile in Arxiv Copilot**. Users can input his/her name and then click "Set Profile" to generate the personalized profile based on historical publication. Besides, if users are unsatisfied with the generated profile or fail to get historical publication, they also can manually edit the profile and then click "Edit Profile".



Get trending topics and ideas!

(1) Input your email: You can sign up with your email and we will send trending research topics, ideas, and papers related to your profile on Monday of every week.

(2) Select time range: We will give you personalized research trend and ideas under selected time range if you have set your profile. Otherwise, general research trend will be provided.

Input your email: yoshua.bengio@mila.quebec

Select time range: ○ day ● week ○ all

Trending Papers

[1] Exploring End-to-end Differentiable Neural Charged Particle Tracking -- A Loss Landscape Perspective: http://arxiv.org/abs/2407.13420v1;
[2] Correlations Are Ruining Your Gradient Descent: http://arxiv.org/abs/2407.10780v1;
[3] Predictive Low Rank Matrix Learning under Partial Observations: Mixed-Projection ADMM: http://arxiv.org/abs/2407.13731v1;
[4] With or Without Replacement? Improving Confidence in Fourier Imaging: http://arxiv.org/abs/2407.13575v1;
[5] A Methodology Establishing Linear Convergence of Adaptive Gradient Methods under PL Inequality: http://arxiv.org/abs/2407.12629v1;
[6] Online Pseudo-Zeroth-Order Training of Neuromorphic Spiking Neural Networks: http://arxiv.org/abs/2407.12516v1;
[7] FSP-Laplace: Function-Space Priors for the Laplace Approximation in Bayesian Deep Learning:

Trending Topics

Here are the top keywords related to the research backgrounds and trends based on the provided paper abstracts:

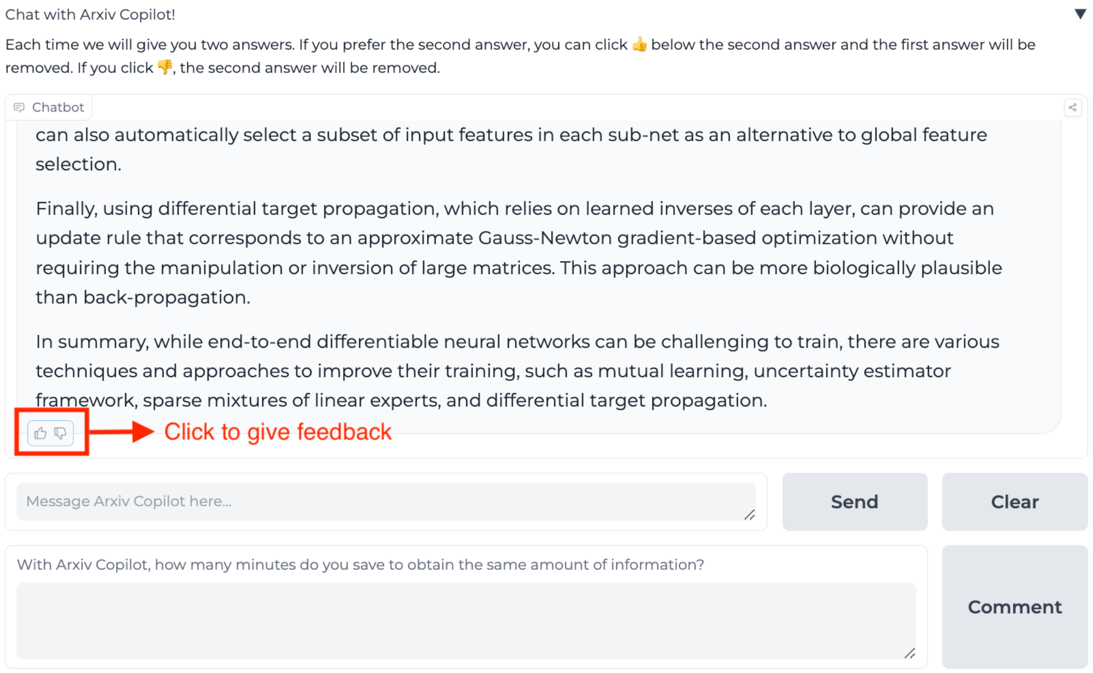
1. End-to-end differentiable neural networks
2. Charged particle tracking
3. Loss landscape perspective
4. Natural gradient descent
5. Data decorrelation
6. Low rank matrix learning
7. Predictive modeling
8. Adaptive gradient methods
9. Neuromorphic computing
10. Uncertainty quantification (UQ)

These keywords represent the focus areas of the papers, including

Ideas for Trending Topics

* End-to-end differentiable neural networks and adaptive gradient methods can be combined to create more efficient and accurate learning algorithms. By making the entire model differentiable, gradients can be calculated and backpropagated through the entire network, allowing for fine-grained optimization of all parameters. Adaptive gradient methods, such as Adam or RMSProp, can further improve convergence and robustness by adjusting the learning rate on a per-parameter basis.
* Neuromorphic computing and low rank matrix learning can be used to create more efficient and scalable machine learning models. Neuromorphic computing, which is inspired by the structure and function of the human brain, can be used to design hardware that is optimized for neural network computations. Low rank matrix learning, on the other hand, can be used to reduce the dimensionality of data and models, which can lead to significant

Figure 10: **Screenshot for the interaction of research trend and ideas in Arxiv Copilot**. Users can sign up with email to receive the weekly update. Besides, users can also select the time range for getting the research trend and we have three choices here *i.e.*day means getting trend from today's papers, week means getting trend from this week's papers and all means getting trend from all papers. After selecting the time range, users can click "Confirm" and the trending papers, trending research topics and ideas will be shown to the users.



Chat with Arxiv Copilot!

Each time we will give you two answers. If you prefer the second answer, you can click 👍 below the second answer and the first answer will be removed. If you click 👎, the second answer will be removed.

Chatbot

can also automatically select a subset of input features in each sub-net as an alternative to global feature selection.

Finally, using differential target propagation, which relies on learned inverses of each layer, can provide an update rule that corresponds to an approximate Gauss-Newton gradient-based optimization without requiring the manipulation or inversion of large matrices. This approach can be more biologically plausible than back-propagation.

In summary, while end-to-end differentiable neural networks can be challenging to train, there are various techniques and approaches to improve their training, such as mutual learning, uncertainty estimator framework, sparse mixtures of linear experts, and differential target propagation.

Click to give feedback

Message Arxiv Copilot here...

With Arxiv Copilot, how many minutes do you save to obtain the same amount of information?

Figure 11: **Screenshot for the interaction of advisory research chat in Arxiv Copilot**. Users can click "send" after entering the question and Arxiv Copilot will give two answers. Specifically, the first answer is with both thought and paper retrieval while the second answer is just with paper retrieval. Here the second answer will have two feedback choices for users, one is 'like' and another is 'dislike'. If users click 'like', the first answer will be removed. Otherwise the second answer will removed. Besides, users can also clean the chat history by clicking "Clear" and provide further feedback by clicking "Comment".