DocAgent: A Research and Writing Assistant for Google Docs

Sahlar Salehi, Akira Durham

May 2025

Abstract

Since the emergence of text language models and LLMs, one of their largest applications has been in the writing domain. However, with many LLM tools there is a lack of user control and collaboration directly with the LLM on the user's working document. This paper proposes DocAgent, a chat-based research and writing agent assistant that can write directly to Google Docs via Google Cloud/Docs API. This allows the user to keep control over their document by making edits directly into the Google Doc, while also enabling optimizations by querying DocAgent to make edits, find and cite sources using SerpAPI web search tools, as well as add new content to the document. DocAgent performs well at finding and citing relevant sources, and integrating the source information into both empty and existing work-in-progress documents. However, currently DocAgent struggles at handling long contexts, and cannot currently apply special formatting to text. Future improvements would involve implementing text formatting and other Google Doc features, as well as improving its ability to handle longer and more complicated queries.

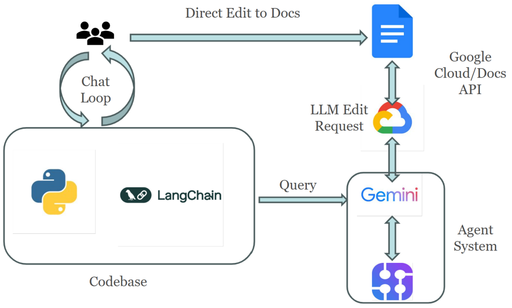


Figure 1: DocAgent general workflow

1 Introduction

With the rise of LLMs, writing and editing text has become a highly optimized task. Chat models like ChatGPT, Claude, and Gemini have enabled users to easily receive text answers to their queries. With these models' ability to perform web searches and research tasks, research writing and source finding has also become a far more trivial task. Tools dedicated to research like Perplexity AI and NotebookLM have expanded upon LLMs abilities to dynamically retrieve information from online sources and provide citations, which provide additional validity to the LLMs responses. These tools also utilize strategies such as RAG to further enhance query-response times and reduce hallucinations [1]. Furthermore, Agent Laboratory is capable of conducting end-to-end research, including literature reviewing, experimentation, and report writing by utilizing a multi-agent approach [2].

However, many LLM tools like ChatGPT and similar chat models struggle with integrating generated text into existing user documents. Firstly, the user is required to dump their document text into the model and paste sections of the LLM output back into their working document, which could be problematic if the LLM provides unacceptable outputs. This makes it difficult to give the user control by requiring them to engage in repetitive querying and pasting parts of the LLM output into their document.

The core problem lies in the lack of a simplified workflow between the user, LLM, and working document which enables the user to maintain full control while also allowing the LLM to provide direct edits based on the user's query as needed. This would reduce the user's need to have unproductive back and forth dialogues with the LLM to try and generate full text that fits the user's requirements. This paper proposes DocAgent, a research and writing agent which can make edits directly to the user's Google Docs, simplifying user and LLM collaboration on document writing.

2 DocAgent

DocAgent is a chat-based Google Docs research and writing agent built using LangChain, powered by Google Gemini 2.0-Flash. At a high level, DocAgent operates as follows: (1) the user provides a document and a query for the agent, (2) the agent calls upon tools as needed and completes the request, (3) the agent outputs the updated document text based on the query, (4) the user can either accept or reject the modifications, and can make another query to repeat the process. In this section, we further describe the workflow in which a user interacts with DocAgent and the agentic system completes a query, the configuration of DocAgent, and its tools to perform research and citation tasks.

2.1 Agentic Workflow

DocAgent's primary role is to write to Google Docs based on the user's request. In the initial prompt to DocAgent, the system is told its core functionality via a system prompt, and is provided the text from the working Google Doc. The system accesses the Google Doc through a Google Cloud Service Account given the document ID and a Google Docs API request. A LangChain agent functions in a feedback loop structure in which the LLM is given an input, and interacts with a set of provided tools based on its action plan to generate an output (Figure 2). DocAgent uses a ReAct agent, which means it reasons on the task to generate actions with specific action inputs [3]. The agent will generate intermediate JSON outputs consisting of an action, which may be the name of the tool to call on, and an input for the action/tool. This provides further insight into the model's thought process, as well as the how the system reasoned about tool use, increasing system interpretability.

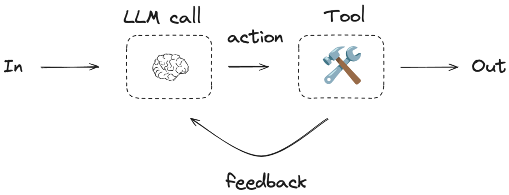
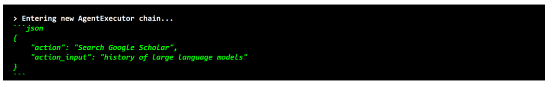


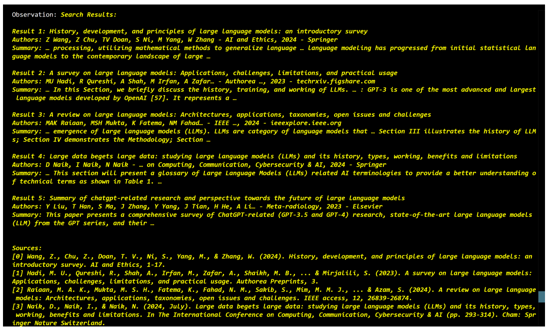
Figure 2: LangChain agent workflow

The user interacts with the DocAgent via a command-line chat loop. When DocAgent receives a user query, it will reason and call upon one of the provided search tools if necessary. If the query involves a simple edit or rewording rather than research, then DocAgent takes this into account and simply uses the attached LLM to make the changes, reducing complexity. In calling search tools, DocAgent will generate the search query as an action input based on the user's prompt (Figure 3). Then, after receiving the tool output (Figure 4), it will generate the final output (Figure 5, 6), which the user can approve to write to the Google Doc via a Docs API request, or deny to reiterate the process. DocAgent is instructed to specifically output the full document text including the edits, new content, and citations made to the document, so it will first send a request to clear the document, and then another to write its output to the document, which includes the original document content to preserve the previous state. Furthermore, the user can make more queries in the chat loop.



~~~~
> Entering new AgentExecutor chain...
```json
{
    "action": "Search Google Scholar",
    "action_input": "history of large language models"
}
```
~~~~

Figure 3: DocAgent selecting Google Scholar search tool as action with generated query as action input



~~~~
Observation: Search Results:

Result 1: History, development, and principles of large language models: an introductory survey
Authors: Z Wang, Z Chu, TV Doan, S Ni, M Yang, W Zhang - AI and Ethics, 2024 - Springer
Summary: … processing, utilizing mathematical methods to generalize language … language modeling has progressed from initial statistical lan
guage models to the contemporary landscape of large …

Result 2: A survey on large language models: Applications, challenges, limitations, and practical usage
Authors: MU Hadi, R Qureshi, A Shah, M Irfan, A Zafar… - Authorea …, 2023 - techrxiv.figshare.com
Summary: … In this Section, we briefly discuss the history, training, and working of LLMs. … : GPT-3 is one of the most advanced and largest
 language models developed by OpenAI [57]. It represents a …

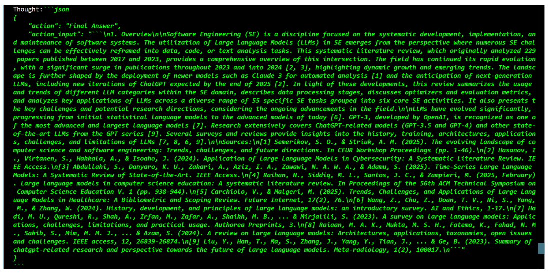
Result 3: A review on large language models: Architectures, applications, taxonomies, open issues and challenges
Authors: MAK Raiaan, MSH Mukta, K Fatema, NM Fahad… - IEEE …, 2024 - ieeexplore.ieee.org
Summary: … emergence of large language models (LLMs). LLMs are category of language models that … Section III illustrates the history of LLM
s; Section IV demonstrates the Methodology; Section …

Result 4: Large data begets large data: studying large language models (LLMs) and its history, types, working, benefits and limitations
Authors: D Naik, I Naik, N Naik - … on Computing, Communication, Cybersecurity & AI, 2024 - Springer
Summary: … This section will present a glossary of Large Language Models (LLMs) related AI terminologies to provide a better understanding o
f technical terms as shown in Table 1. …

Result 5: Summary of chatgpt-related research and perspective towards the future of large language models
Authors: Y Liu, T Han, S Ma, J Zhang, Y Yang, J Tian, H He, A Li… - Meta-radiology, 2023 - Elsevier
Summary: This paper presents a comprehensive survey of ChatGPT-related (GPT-3.5 and GPT-4) research, state-of-the-art large language models
(LLM) from the GPT series, and their …

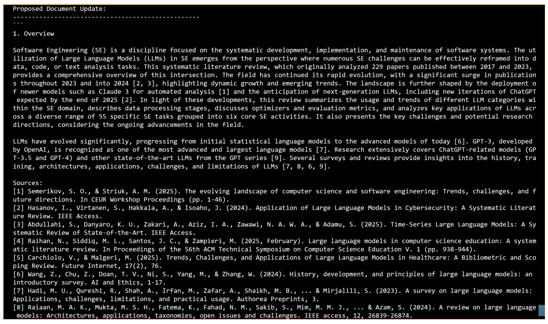

Sources:
[0] Wang, Z., Chu, Z., Doan, T. V., Ni, S., Yang, M., & Zhang, W. (2024). History, development, and principles of large language models: an
introductory survey. AI and Ethics, 1-17.
[1] Hadi, M. U., Qureshi, R., Shah, A., Irfan, M., Zafar, A., Shaikh, M. B., ... & Mirjalili, S. (2023). A survey on large language models:
Applications, challenges, limitations, and practical usage. Authorea Preprints, 3.
[2] Raiaan, M. A. K., Mukta, M. S. H., Fatema, K., Fahad, N. M., Sakib, S., Mim, M. M. J., ... & Azam, S. (2024). A review on large language
 models: Architectures, applications, taxonomies, open issues and challenges. IEEE access, 12, 26839-26874.
[3] Naik, D., Naik, I., & Naik, N. (2024, July). Large data begets large data: studying large language models (LLMs) and its history, types,
 working, benefits and limitations. In The International Conference on Computing, Communication, Cybersecurity & AI (pp. 293-314). Cham: Spr
inger Nature Switzerland.
~~~~

Figure 4: DocAgent Google Scholar search tool result



~~~~
Thought:```json
{
    "action": "Final Answer",
    "action_input": "```\n1. Overview\n\nSoftware Engineering (SE) is a discipline focused on the systematic development, implementation, an
d maintenance of software systems. The utilization of Large Language Models (LLMs) in SE emerges from the perspective where numerous SE chal
lenges can be effectively reframed into data, code, or text analysis tasks. This systematic literature review, which originally analyzed 229
 papers published between 2017 and 2023, provides a comprehensive overview of this intersection. The field has continued its rapid evolution
, with a significant surge in publications throughout 2023 and into 2024 [2, 3], highlighting dynamic growth and emerging trends. The landsc
ape is further shaped by the deployment of newer models such as Claude 3 for automated analysis [1] and the anticipation of next-generation
LLMs, including new iterations of ChatGPT expected by the end of 2025 [2]. In light of these developments, this review summarizes the usage
and trends of different LLM categories within the SE domain, describes data processing stages, discusses optimizers and evaluation metrics,
and analyzes key applications of LLMs across a diverse range of 55 specific SE tasks grouped into six core SE activities. It also presents t
he key challenges and potential research directions, considering the ongoing advancements in the field.\n\nLLMs have evolved significantly,
progressing from initial statistical language models to the advanced models of today [6]. GPT-3, developed by OpenAI, is recognized as one o
f the most advanced and largest language models [7]. Research extensively covers ChatGPT-related models (GPT-3.5 and GPT-4) and other state-
of-the-art LLMs from the GPT series [9]. Several surveys and reviews provide insights into the history, training, architectures, application
s, challenges, and limitations of LLMs [7, 8, 6, 9].\n\nSources:\n[1] Semerikov, S. O., & Striuk, A. M. (2025). The evolving landscape of co
mputer science and software engineering: Trends, challenges, and future directions. In CEUR Workshop Proceedings (pp. 1-46).\n[2] Hasanov, I
., Virtanen, S., Hakkala, A., & Isoaho, J. (2024). Application of Large Language Models in Cybersecurity: A Systematic Literature Review. IE
EE Access.\n[3] Abdullahi, S., Danyaro, K. U., Zakari, A., Aziz, I. A., Zawawi, N. A. W. A., & Adamu, S. (2025). Time-Series Large Language
Models: A Systematic Review of State-of-the-Art. IEEE Access.\n[4] Raihan, N., Siddiq, M. L., Santos, J. C., & Zampieri, M. (2025, February)
. Large language models in computer science education: A systematic literature review. In Proceedings of the 56th ACM Technical Symposium on
 Computer Science Education V. 1 (pp. 938-944).\n[5] Carchiolo, V., & Malgeri, M. (2025). Trends, Challenges, and Applications of Large Lang
uage Models in Healthcare: A Bibliometric and Scoping Review. Future Internet, 17(2), 76.\n[6] Wang, Z., Chu, Z., Doan, T. V., Ni, S., Yang,
 M., & Zhang, W. (2024). History, development, and principles of large language models: an introductory survey. AI and Ethics, 1-17.\n[7] Ha
di, M. U., Qureshi, R., Shah, A., Irfan, M., Zafar, A., Shaikh, M. B., ... & Mirjalili, S. (2023). A survey on large language models: Applic
ations, challenges, limitations, and practical usage. Authorea Preprints, 3.\n[8] Raiaan, M. A. K., Mukta, M. S. H., Fatema, K., Fahad, N. M
., Sakib, S., Mim, M. M. J., ... & Azam, S. (2024). A review on large language models: Architectures, applications, taxonomies, open issues
and challenges. IEEE access, 12, 26839-26874.\n[9] Liu, Y., Han, T., Ma, S., Zhang, J., Yang, Y., Tian, J., ... & Ge, B. (2023). Summary of
chatgpt-related research and perspective towards the future of large language models. Meta-radiology, 1(2), 100017.\n```"
}
```
~~~~

Figure 5: DocAgent output as thought process



~~~~
Proposed Document Update:
--------------------------------------------------
```
1. Overview

Software Engineering (SE) is a discipline focused on the systematic development, implementation, and maintenance of software systems. The ut
ilization of Large Language Models (LLMs) in SE emerges from the perspective where numerous SE challenges can be effectively reframed into d
ata, code, or text analysis tasks. This systematic literature review, which originally analyzed 229 papers published between 2017 and 2023,
provides a comprehensive overview of this intersection. The field has continued its rapid evolution, with a significant surge in publication
s throughout 2023 and into 2024 [2, 3], highlighting dynamic growth and emerging trends. The landscape is further shaped by the deployment o
f newer models such as Claude 3 for automated analysis [1] and the anticipation of next-generation LLMs, including new iterations of ChatGPT
 expected by the end of 2025 [2]. In light of these developments, this review summarizes the usage and trends of different LLM categories wi
thin the SE domain, describes data processing stages, discusses optimizers and evaluation metrics, and analyzes key applications of LLMs acr
oss a diverse range of 55 specific SE tasks grouped into six core SE activities. It also presents the key challenges and potential research
directions, considering the ongoing advancements in the field.

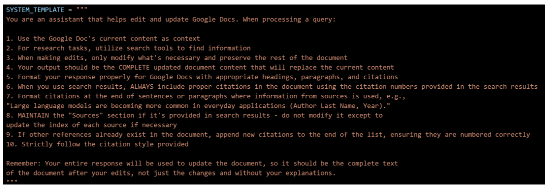
LLMs have evolved significantly, progressing from initial statistical language models to the advanced models of today [6]. GPT-3, developed
by OpenAI, is recognized as one of the most advanced and largest language models [7]. Research extensively covers ChatGPT-related models (GP
T-3.5 and GPT-4) and other state-of-the-art LLMs from the GPT series [9]. Several surveys and reviews provide insights into the history, tra
ining, architectures, applications, challenges, and limitations of LLMs [7, 8, 6, 9].

Sources:
[1] Semerikov, S. O., & Striuk, A. M. (2025). The evolving landscape of computer science and software engineering: Trends, challenges, and f
uture directions. In CEUR Workshop Proceedings (pp. 1-46).
[2] Hasanov, I., Virtanen, S., Hakkala, A., & Isoaho, J. (2024). Application of Large Language Models in Cybersecurity: A Systematic Literat
ure Review. IEEE Access.
[3] Abdullahi, S., Danyaro, K. U., Zakari, A., Aziz, I. A., Zawawi, N. A. W. A., & Adamu, S. (2025). Time-Series Large Language Models: A Sy
stematic Review of State-of-the-Art. IEEE Access.
[4] Raihan, N., Siddiq, M. L., Santos, J. C., & Zampieri, M. (2025, February). Large language models in computer science education: A system
atic literature review. In Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1 (pp. 938-944).
[5] Carchiolo, V., & Malgeri, M. (2025). Trends, Challenges, and Applications of Large Language Models in Healthcare: A Bibliometric and Sco
ping Review. Future Internet, 17(2), 76.
[6] Wang, Z., Chu, Z., Doan, T. V., Ni, S., Yang, M., & Zhang, W. (2024). History, development, and principles of large language models: an
introductory survey. AI and Ethics, 1-17.
[7] Hadi, M. U., Qureshi, R., Shah, A., Irfan, M., Zafar, A., Shaikh, M. B., ... & Mirjalili, S. (2023). A survey on large language models:
Applications, challenges, limitations, and practical usage. Authorea Preprints, 3.
[8] Raiaan, M. A. K., Mukta, M. S. H., Fatema, K., Fahad, N. M., Sakib, S., Mim, M. M. J., ... & Azam, S. (2024). A review on large language
 models: Architectures, applications, taxonomies, open issues and challenges. IEEE access, 12, 26839-26874.
~~~~

Figure 6: DocAgent final output (user given option to accept below the shown text)

2.2 Configuration

We configure DocAgent through the specification of LangChain agents. LangChain agents consist of an agent type specification, an LLM, and a set of custom-curated LLM callable tools. DocAgent is a LangChain Conversational ReAct agent, which utilizes ReAct prinicples to improve task action planning, and is optimized for conversations. For its LLM, DocAgent uses Google Gemini 2.0-Flash, which was selected due to its high performance among free-use models. We provided Gemini with a custom system prompt (Figure 7). The system prompt was designed to inform the model of its use as an editing and research assistant, reinforcing that it should output the full document texts with edits made to only the necessary components based on the query, and to include and maintain citations strictly following a specified format (APA, MLA, etc). We further provide DocAgent with a conversation memory so it can access previous messages and make corrections or continuations of previous ideas based on the context.



```
SYSTEM_TEMPLATE = """
You are an assistant that helps edit and update Google Docs. When processing a query:

1. Use the Google Doc's current content as context
2. For research tasks, utilize search tools to find information
3. When making edits, only modify what's necessary and preserve the rest of the document
4. Your output should be the COMPLETE updated document content that will replace the current content
5. Format your response properly for Google Docs with appropriate headings, paragraphs, and citations
6. When you use search results, ALWAYS include proper citations in the document using the citation numbers provided in the search results
7. Format citations at the end of sentences or paragraphs where information from sources is used, e.g.,
"Large language models are becoming more common in everyday applications (Author Last Name, Year)."
8. MAINTAIN the "Sources" section if it's provided in search results - do not modify it except to
update the index of each source if necessary
9. If other references already exist in the document, append new citations to the end of the list, ensuring they are numbered correctly
10. Strictly follow the citation style provided

Remember: Your entire response will be used to update the document, so it should be the complete text
of the document after your edits, not just the changes and without your explanations.
"""
```

Figure 7: DocAgent Gemini system prompt

2.3 Research Tools

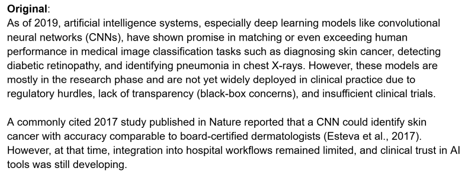
DocAgent is equipped with two search tools which leverage SerpAPI, a popular web-search API, to assist with research and citation tasks: a general web search, and a Google Scholar web search. DocAgent is able to call upon these tools based on the generated action in its action plan, and provides a query for the search as an action input (Figure 3). The regular web search tool calls upon SerpAPI and retrieves the information, link, and title of the top five sources for the provided query, utilizing a standard Google search. The tool outputs a list of summarized content from the sources followed by a list of the content titles and links to the sources. The Google Scholar web search is focused for research oriented tasks and specific style citations, which makes specific Google Scholar searches. Upon entering the DocAgent chat loop, the user specifies a citation style of either APA, MLA, Chicago, Harvard, or Vancouver. The Google Scholar web search tool will provide a list of content retrieved from the top five Google Scholar sources followed by a list of citations in the specified format for those sources, which is pulled directly from the Google Scholar provided citation (Figure 4).

3 Results

We evaluate DocAgent through rudimentary tests on generating and editing texts, as well as on researching and citing sources while integrating this new information into working documents. Then we discuss some key limitations of DocAgent.

3.1 Testing

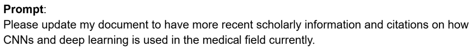
In testing DocAgent's research and writing abilities, we focused primarily on its ability to work on existing documents. In one test, we provided DocAgent with a brief text on convolutional neural networks (CNN) in the medical field (Figure 8). We prompted DocAgent to update the document with more recent information on how CNNs are used in the medical field with sources (Figure 9). Based on the prompt and the document, DocAgent generated an output (Figure 10) which includes researched information with in-text citation indices, relevant parts of the original text, and edits to recontextualize the original document to integrate the researched information. Furthermore, a list of references was provided at the end of the document in APA format (Figure 11). We conducted further tests, including simple edits as well as research-orientated requests. A few examples of DocAgent testing can be seen in the link provided, which details 4 ”common-use” situations where we provide the initial text, the prompt, and the model output based on the text and prompt. Google Doc with Examples



Original:
As of 2019, artificial intelligence systems, especially deep learning models like convolutional neural networks (CNNs), have shown promise in matching or even exceeding human performance in medical image classification tasks such as diagnosing skin cancer, detecting diabetic retinopathy, and identifying pneumonia in chest X-rays. However, these models are mostly in the research phase and are not yet widely deployed in clinical practice due to regulatory hurdles, lack of transparency (black-box concerns), and insufficient clinical trials.

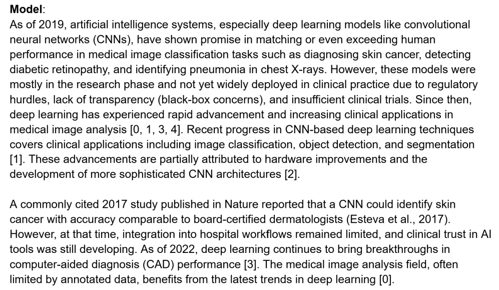
A commonly cited 2017 study published in Nature reported that a CNN could identify skin cancer with accuracy comparable to board-certified dermatologists (Esteva et al., 2017). However, at that time, integration into hospital workflows remained limited, and clinical trust in AI tools was still developing.

Figure 8: Sample document input for DocAgent about CNNs in medicine



Prompt:
Please update my document to have more recent scholarly information and citations on how CNNs and deep learning is used in the medical field currently.

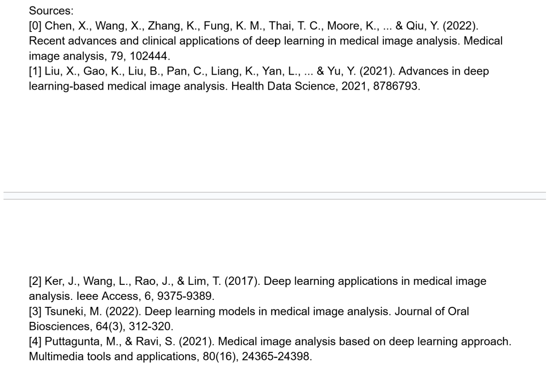
Figure 9: User prompt to DocAgent for document shown in Figure 8



Model:
As of 2019, artificial intelligence systems, especially deep learning models like convolutional neural networks (CNNs), have shown promise in matching or even exceeding human performance in medical image classification tasks such as diagnosing skin cancer, detecting diabetic retinopathy, and identifying pneumonia in chest X-rays. However, these models were mostly in the research phase and not yet widely deployed in clinical practice due to regulatory hurdles, lack of transparency (black-box concerns), and insufficient clinical trials. Since then, deep learning has experienced rapid advancement and increasing clinical applications in medical image analysis [0, 1, 3, 4]. Recent progress in CNN-based deep learning techniques covers clinical applications including image classification, object detection, and segmentation [1]. These advancements are partially attributed to hardware improvements and the development of more sophisticated CNN architectures [2].

A commonly cited 2017 study published in Nature reported that a CNN could identify skin cancer with accuracy comparable to board-certified dermatologists (Esteva et al., 2017). However, at that time, integration into hospital workflows remained limited, and clinical trust in AI tools was still developing. As of 2022, deep learning continues to bring breakthroughs in computer-aided diagnosis (CAD) performance [3]. The medical image analysis field, often limited by annotated data, benefits from the latest trends in deep learning [0].

Figure 10: DocAgent output to document in Figure 8 given the prompt in Figure 9



Sources:
[0] Chen, X., Wang, X., Zhang, K., Fung, K. M., Thai, T. C., Moore, K., ... & Qiu, Y. (2022). Recent advances and clinical applications of deep learning in medical image analysis. Medical image analysis, 79, 102444.
[1] Liu, X., Gao, K., Liu, B., Pan, C., Liang, K., Yan, L., ... & Yu, Y. (2021). Advances in deep learning-based medical image analysis. Health Data Science, 2021, 8786793.

[2] Ker, J., Wang, L., Rao, J., & Lim, T. (2017). Deep learning applications in medical image analysis. Ieee Access, 6, 9375-9389.
[3] Tsuneki, M. (2022). Deep learning models in medical image analysis. Journal of Oral Biosciences, 64(3), 312-320.
[4] Puttagunta, M., & Ravi, S. (2021). Medical image analysis based on deep learning approach. Multimedia tools and applications, 80(16), 24365-24398.

Figure 11: DocAgent output citations associated with the output in Figure 10

3.2 Limitations

In the above example, among others, DocAgent succeeds in retrieving and integrating relevant external sources with its search tools, as well as providing edits and citations in a requested style to the document while maintaining key parts from the original text. However, there are key limitations to DocAgent's capabilities.

Firstly, in its current implementation, DocAgent uses a free-tier Google Gemini model, which may underperform compared to more powerful paid-tier models in terms of response time, model capability, and usage limits. We did experiment with other models, however due to the free-tier restrictions, we decided to settle with the most common free-tier Gemini model, Gemini 2.0-Flash. Further development can experiment with DocAgent quality with different LLM families as well as better models for increased context range and text handling.

When DocAgent creates its action plan and calls upon research tools, it far more frequently calls upon the Google Scholar research tool as opposed to the general search tool. This may be a result of the prompt template provided to the agent which is based on a keyword match over the query. While it is often preferable for DocAgent's research purposes to call upon the Google Scholar search tool, it limits the types of sources selected by the SerpAPI web search, potentially leading to an academic focus, rather than relevant but non-academic sources. This could be addressed through more aware keyword searches or using both searches and then fine tune from the returned sources for most relevant.

DocAgent is currently unable to apply text styles (bold, italics, etc) to the Google Docs. This is because the requests used to update the Google Doc are statically defined, not generated by the LLM. DocAgent only provides the updated document text as an input for the Google Docs API request. While DocAgent can provide markdown style text formatting, applying the text formatting in Google Docs would have to be done manually by the user. This limitation also applies to other Google Docs features which require custom API requests to be selected depending on the query to DocAgent. We have thought about applying another agent tool to dynamically create this API request structure to include text styles, however experimentation led to consistent malformed text, causing us to abandon this avenue and focus on other aspects.

4 Conclusion

Existing text-editing LLMs lack seamless integration with user writing environments, creating inefficiencies in the writing workflow. Current solutions often require users to switch to standalone web applications, disrupting the existing digital writing process. To address this, we proposed DocAgent, a writing assistant designed to bridge the gap between native text editors and LLM-based tools.

DocAgent supports both editing and research tasks by utilizing native LLM capabilities and SerpAPI, respectively. Built with LangChain's ReAct agent framework, our agent combines these tools into a more cohesive experience that aims to minimize the friction for its users. DocAgent can locate credible sources, refine text, and integrate new content into the document with user approval at each step, demonstrating its potential to streamline and improve the digital writing process. By aligning AI capabilities with familiar writing environments, DocAgent lays the groundwork for more intuitive and accessible writing tools. Future improvements to DocAgent could include implementing text formatting support, and including more agents to the system to handle a wider range of tasks.

References

[1] Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., ... & Wang, H. (2023). Retrieval-augmented generation for large language models: A survey. arXiv preprint arXiv:2312.10997, 2, 1.

[2] Schmidgall, S., Su, Y., Wang, Z., Sun, X., Wu, J., Yu, X., ... & Barsoum, E. (2025). Agent laboratory: Using llm agents as research assistants. arXiv preprint arXiv:2501.04227.

[3] Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., & Cao, Y. (2023, January). React: Synergizing reasoning and acting in language models. In International Conference on Learning Representations (ICLR).