# Are LLMs better at using search engines than humans?

Author: Nils Klüwer, 12229263
e12229263@student.tuwien.ac.at

[1] Vienna University of Technology, Vienna, Austria
[2] https://www.tuwien.at/en/
[3] Institute of Information Systems Engineering

**Abstract.** This report explores whether Large Language Models (LLMs) can outperform humans in using search engines to retrieve information. Using the DuckDuckGo API and the GPT-3.5 model from OpenAI, we compare two approaches: a simple method using the user initial query and an augmented method where the LLM generates alternative queries. Through an evaluation employing questions of varying complexity, we find that the augmented method significantly improves relevance, coherence, and completeness in complex queries, while showing minimal benefit for straightforward ones. This study underscores the potential of LLMs to enhance search engine effectiveness by diversifying search inputs and providing more comprehensive results. Challenges such as resource consumption and the need for robust evaluation methods are also discussed. The findings point toward promising future applications of LLMs in information retrieval.

**Keywords:** DuckDuckGo · LLM · Information Search · LLM supported Information Retrieval.

## 1 Introduction

The recent advancements of technology in the field of artificial intelligence are Large Language Models. It was questioned by some researchers, shortly after the release of ChatGPT with later studies of comparing the performance of google vs. ChatGPT. Xu et. al. findings where that ChatGPT notably "levels user search performance across different education levels and excels in answering straightforward questions and providing general solutions but falls short in fact-checking tasks."[XFC23, p.1] The consequence of these limitations was to give the model some form of grounding. This is done by enabling ChatGPT to perform information retrieval via a search engine (Microsoft Bing). OpenAi released the "Web Browsing" feature for ChatGPT on May 12, 2023. It enabled the model to trigger a Web search. Since OpenAi is no longer "open" as the name suggest, but closed source it is not possible to know for sure how they implemented their "Web browsing" feature. In this work two methods that potentially improve the quality and relevance of search results are compared, the simple vs. augmented approach. [XFC23; Ope23]

This investigation of methods closely aligns with the course "Informationssuche im Internet" focusing on the enhancement of information retrieval through the use of APIs and state of the art technologies. In the report technical details are explained, the results, drawbacks and limitations as well as effectiveness of such a solution.

## 2 Methodology

The objective of the methodology part is to give a clear understanding of how the APIs and mix of these architecture was leveraged to create a system that could potentially outperform simple search approaches.

The overall framework for this project is based on design science, producing a Python program that utilizes APIs and a combination of architectural designs.

### 2.1 The APIs

**DuckDuckGo:** There are simple reasons why the DuckDuckGo API was chosen, it is easy to use and it is free to use. Both things which were not true for the Google search engine, which is mostly hidden behind a paywall. The code is written in python where the "duckduckgo-search 5.3.0" python package is used. The text search of the API lets you input a number of parameter: keywords, region, safesearch, max_results. These are standard parameters, the 'keywords' equals the user search query, the region lets you select specific regions like 'de-de' for the search region Germany with output in German language or 'uk-en' for United Kingdom with English language or 'wt-wt' for World with the default language (which is in english). The 'safesearch' controls the level of filtering for adult content. The 'max_result' is important to set a fixed number of results. How many results should be returned for the given query. In the setup of the study the safesearch is set 'moderate', the language is set to 'wt-wt' and the max_results to 10 for the simple search, 2 for the augmented search. The augmented search has the option to make up to 5 queries with each receiving the top 2 results, so both approaches can receive up to 10 results in total. The response of the API contains a 'body', 'title' and 'URL' which were used to get insight into the results of the search engine. The specific sites were not investigated any further. The output, and possible text to ground the language model and therefore augment the search is heavily dependent on the 'body' and the 'title' of the search result. The 'body' is the snippet which is shown below a search results and contains approximately between 50-75. Limitations of this approach and usage of that API without retrieving the full content of a search results website is discussed later. In 1.1 the basic API call can be seen, which is straight forward and easy to use.

```
1 from duckduckgo_search import DDGS
2
```

```
def duckduckgo_text_search(keywords, region, safesearch,
    timelimit, max_results):
    ddgs = DDGS()
    results = ddgs.text(
        keywords=keywords,
        region=region,
        safesearch=safesearch,
        timelimit=timelimit,
        max_results=max_results,
    )
    return results
```

**Listing 1.1.** Python code for basic DuckDuckGo API call

**OpenAi:** There are many offerings of models to use. To keep it short, OpenAi is the current marked leader and the Company who made this topic big. Their models are affordable and easy to use. The documentation is complete, with many examples offering an easy entry into the topic. The model can of course be exchanged with open source models. But since this is not the focus of this work, exploring models and possible open source solutions, the easiest (subjective) API providing function calling with a low latency and long term support was chosen. In 1.2 the basic chat completion API call is shown, which is used to call the model. The model chosen is "gpt-3.5-turbo-0125", is the cheapest model which has the capability to function call. The message is the input which we constructed via the system prompt and user message. The "tool_choice" is set to "required" to force the model to call a function. The alternative of setting it to "auto" works as well. There was no instance during this project were "auto" did not work, best practise although should be to make it explicit and provide as many rail guards as possible. These rail guards can improve the robustness and reliability of the model to perform the desired action. This becomes more relevant if multiple functions are provided with more complex tasks to solve. The parameter "temperature" is set to 0.0 to ensure that the model will pick the most probable tokens when generating the response. This can improves reproducibility and will produce more similar responses. In the end the responses are not deterministic, only through the usage of "seeds" this could be archived. The parameters for all model request are the same.

```
response = client.chat.completions.create(
        model="gpt-3.5-turbo-0125",
        temperature=0.0,
        messages=messages,
        tools=tools,
        tool_choice="required"
    )
```

**Listing 1.2.** Python code for basic OpenAi API call

### 2.2 System Architecture

It is evaluated if the model is capable of outperforming human made search queries. In this setup the DuckDuckGo API is used, in combination with the ChatGPT3.5 API. In 1 both approaches can be seen. It is important to mention that the underlying function, making the DuckDuckGo search engine request is the same. The difference lies in what is passed to the function. In the simple approach the raw user query is used, in the augmented approach GPT-3.5 generates up to five queries which are then passed to the DuckDuckGO API. GPT-3.5 is instructed to generate good queries to find the best answers when asking a search engine. A job the human does normally by typing his search query / keywords into a search engine.

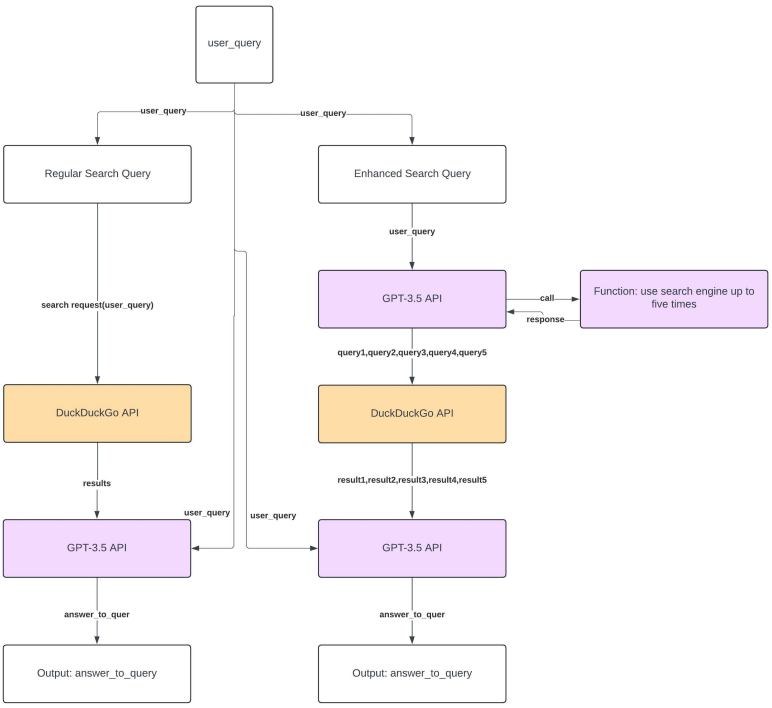


**Fig. 1.** Flow of how the user_query is processed by both approaches of using Web browsing.

### 2.3 Prompting and Tool usage

The prompts used are crucial to understand how the summary in the end is created and how the model is conducting the search in the augmented approach.

There are two prompts in total used, the summarization prompt and the prompt to trigger the tool use in the augmented approach.

```
system_message = """Du bist ein search query / keywords expert
    . Du musst 5 Suchanfragen durchfuehren. Die fuenf
    suchanfragen sollten so aufgebaut sein, das es dir gelingt
     eine moeglichst umfassenden Antwort zu bekommen fuer die
    gegeben Suchanfrage. Nutze dafuer die Tools des function
    calling und nutze die moeglichkeit Keywords mit operatoren
     zu veraendern.
     Folgenden optionen gibt es:
        Du kannst verschiedene Suchoperatoren verwenden, um
    deine Suche zu verfeinern:
        1. 'Katzen Hunde' - Ergebnisse ueber Katzen oder Hunde
    .
        2. \'Katzen und Hunde\' - Ergebnisse fuer den genauen
    Begriff 'Katzen und Hunde'.
        3. 'Katzen -Hunde' - Weniger Hunde in den Ergebnissen.
        4. 'Katzen +Hunde' - Mehr Hunde in den Ergebnissen.
        WICHTIG: Du bekommst einen Strafpunkt, sollten gleiche
     Ergebnisse bei dem Ergebnisse der Suche rauskommen.
    """
```

**Listing 1.3.** Promtp instructing the model to generate alternative queries

In 1.3 is one part of the messages send to the model. The other relevant part of the message is the "tool" itself, where the model gets the instructions on how to use the tool and what parameters the model can change. This structure is shown in 1.6. These are the information's the model gets to conduct up to five search requests. The instructions direct the model to conduct a search, it is made aware of the fact that it can construct up to five queries and that it will get a penalty. Which is a strategy to force the model into more diverse answers and reduce laziness.

The summarization prompt shown in 1.4 is for both approaches the same and produces the final output which is compared. Since prompt engineering is a new topic and things change quickly regarding the formation and writing of the prompt, only a few things which are already established will be mentioned. The use of "step by step" is intended to trigger a "chain of thought". The model receives guidance on which steps to perform to simulate reasoning. The model cannot reason, it can at most follow a structure and generate the most probable tokens leading to an answer. The current best practise is to conduct the websites of the model provider when it comes to questions regarding prompting. In this case it is OpenAi, other model provider such as google or Anthropic are also giving other advice on prompting the model. Other possible strategies can be found in blogs, it is important to mention that many strategies are very use case specific. There are only a few tactics which are generally applicable. [Ope; Teo24]

It can be discussed on the improvements of these prompt, but since this is not focus of this work, a simple prompting was chosen.

```
system_message = f"""Beantworte nun die urspruengliche Frage
    oder Query step by step. Begruende mit Quellen wie zu dem
    Ergebnis kommst, gebe deine Quellen vollstaendig mit URL
    an! Und schlage die am relevantesten Webseiten seiten vor!
{keywords}"""
```

**Listing 1.4.** Promtp instructing the model to summarize

## 3 Evaluation and Results

In this section it is first explained what is evaluated, then the results are presented. It is clear that easy questions will yield good answers. The model can answer them with its general knowledge, excpet they are time dependent and occured after the training of the model. We therefore move from easy questions up to complex questions. Complexity is in this context referred to a question which will not yield in a high Precision due to the sponginess of the query, like : "What is the meaning of life".

```
queries = [
    # Easy Questions
    "What is the capital of Italy?",
    "Best laptops for students 2023",

    # Moderate Complexity Questions
    "Details of the Treaty of Versailles",
    "Results of the 2022 FIFA World Cup final",
    "How does blockchain technology work?",

    # Complex Questions
    "What are the pros and cons of keto diet?",
    "Impact of Brexit on UK economy",
    "Which is better for home use: solar panels or wind
    turbines?",

    # Highly Complex and Ambiguous Questions
    "Is artificial intelligence beneficial to society?",
    "What is the best way to achieve a work-life balance?"
]
```

**Listing 1.5.** Evaluation questions used

The queries were generated by GPT-4o, the newest model of OpenAi, used prompt are in the Appendix 1.7. We will also use this model to evaluate the results. This is an upcoming topic in the field of LLM´s. Who is evaluating the huge sets of data? This method is used here without further in detail explanation, but the code can be found in the Codebase. The prompt used for evaluation is inspired by the approach suggested by Beurer-Kellner, Luca, et al. [Beu+23; Liu+23; LMQ].

**Evaluation short explained:** The evaluation is done on the summaries generated based on both approaches. The summary of the augmented appraoch vs. the summary of the regular approach are compared by the evaluator LLM. The model is provided with the user query, and both results (A = augmented, B = regular) , seen in 1.8. The model is instructed to write a reasoning statement on why A or B is better, after that statement the model awards a score. This process is done 5 times for each result to ensure consistency, and show that the model will get to similar results each time. The Scores are Coherence, Relevance and Completeness. Coherence is the Logical consistency and fluency of text. Relevance is how well the responses address the query specific information need. Completeness, if the response cover all the essential information required to adequately answer the query.

## 4 Results

We look at two different results, the Scores and evaluation done by the model and the enhanced queries generated by the model, and if they are really enhanced. This will give insight into how the model handled the opportunity to generate alternate queries. These alternative queries compared to the initial user queries shows what strategy the model "thinks" is best to get the most relevant resulsts from the search engine. For the queries in 1.5 the results are the following:

| **Initial and Augmented Queries** |
|---|
| 1. What is the capital of Italy?<br><br>– Capital of Italy<br>– Capital Italy<br>– What is the capital city of Italy<br>– Rome capital Italy<br>– Italy capital city |
| 2. Best laptops for students 2023<br><br>– Best laptops for students 2023<br>– Best laptops +students 2023<br>– Best laptops -students 2023<br>– "Best laptops for students 2023"<br>– Best +laptops +for +students +2023 |
| 3. Details of the Treaty of Versailles<br><br>– Details of the Treaty of Versailles<br>– "Terms of the Treaty of Versailles"<br>– Treaty of Versailles -impact<br>– Treaty of Versailles +reparations<br>– "Treaty of Versailles" +Germany |

| **Initial and Augmented Queries** |
|---|
| 4. Results of the 2022 FIFA World Cup final<br><br>– Results of the 2022 FIFA World Cup final<br>– 2022 FIFA World Cup final winner<br>– 2022 FIFA World Cup final score<br>– 2022 FIFA World Cup final highlights<br>– 2022 FIFA World Cup final best player |
| 5. How does blockchain technology work?<br><br>– How does blockchain technology work?<br>– How does blockchain technology function?<br>– How does blockchain technology operate?<br>– Explanation of blockchain technology workings<br>– Blockchain technology mechanism explained |
| 6. What are the pros and cons of keto diet?<br><br>– pros and cons of ketogenic diet<br>– benefits of keto diet<br>– risks of ketogenic diet<br>– keto diet success stories<br>– keto diet vs other diets |
| 7. Impact of Brexit on UK economy<br><br>– Impact of Brexit on UK economy<br>– Impact of Brexit on UK economy -consequences<br>– Brexit economic impact UK<br>– Brexit impact on British economy<br>– Brexit effects on UK economy |
| 8. Which is better for home use: solar panels or wind turbines?<br><br>– solar panels for home use<br>– wind turbines for residential use<br>– solar panels vs wind turbines for home energy<br>– benefits of solar panels for residential properties<br>– advantages of wind turbines for home use |
| 9. Is artificial intelligence beneficial to society?<br><br>– benefits of artificial intelligence to society<br>– positive impact of artificial intelligence on society<br>– advantages of artificial intelligence for society<br>– how artificial intelligence benefits society<br>– role of artificial intelligence in societal development |
| 10. What is the best way to achieve a work-life balance?<br><br>– best practices for achieving work-life balance<br>– tips for maintaining work-life balance<br>– strategies for balancing work and personal life<br>– importance of work-life balance<br>– impact of work-life balance on mental health |

**Table 1.** Initial Queries and Their Augmented Variants

**Easy Questions:** "What is the capital of Italy?" the model suggests using simpler and more concrete search phrases like "Capital Italy". The usage of boolean operators or "" also shows that the model is capable of using this syntax to refine the search query.

**Moderate Complexity Questions:** In question on "Details of the Treaty of Versailles" and "Results of the 2022 FIFA World Cup final", the model suggests searching for different aspects of a topic. In the 4th questions the augmented queries 2-5, are all almost the same except these words: "winner, score, highlights, best player". This can lead to much more specific outcomes, compared to the initial question.

**Complex Questions:** In these scenarios like "Which is better for home use: solar panels or wind turbines?" the model again cuts a lot of fill words and altering through words. All queries are different, there no such similarity as seen in questions 5,4,2,1. The more complex the more diverse in language and grammar the augmented suggestes queries get. This suggests that this method can leverage a higher potential in more complex questions, compared to trivial ones.

**Highly Complex and Ambiguous Questions:** The same pattern as before can be observed, to pick one interesting augmented query: "importance of work-life balance", this goes a step further compared to just changing the wording of the initial query. This could also potentially return a results which questions "work-life balance" importance itself. This would enable the output to be critical towards the question itself. The query would help the user to break out of an echo chamber by introducing different viewpoints of the topic which are outside of the current base assumptions, which is: "work-life balance is important".

There is much more possible interpretation, but to keep it short, the output which is shown here could be much different when changing the model, the temperature and the system prompts. This is just one instance of a possible outcome, which shows how the model "works" and what to expect when with working with such a solutions.

### 4.1 Evaluation of Final Answers

The full final answers of all ten queries can be found in the GitHub repository in the folder "./output/evaluations_results.json.". There are all 20 Answers, resulting in 445 words. As explained in the methodology part a more advanced model was used to evaluate the results. All 20 Answer were taken, put into the template shown in 1.8 and were compared. This was done for every result 5 times. In total this resulted in 50 evaluation results seen in 2. The "MEAN" is calculated as well es the "MEAN DIFF" which shows the difference between Augmented and Regulars respective score for Coherence, Relevance and Completeness. The "OVERALL DIFF" metric accumulates each "MEAN DIFF" of a

column. The results show that there is at strong indication that the Augmented approach outperforms the regular approach. Only in one case, questions 5 the regular approach beats the Augmented approach by a small margin. All other summaries done by the augmented approach answering the initial question outperform the regular approach according to the LLM evaluation. The OVERALL DIFF especially in Coherence with 8.6 and Completenss with 10.8 are showing that the summary produced based of the augmented approach are better in the evaluated parameters. The more complex the questions, the higher the "DIFF" between regular and augmented approach.

## 5 Discussion

Augmenting and grounding language models is already becoming a common practise to reduce hallucinations, shown by Google , Microsoft, Cohere. RAG - Retrieval-Augmented Generation is based on the same principle, enriching the context of the language models context window to increase quality and reduce hallucination and add information which is not accessible via the general knowledge of the model. This work is grounding the LLM with knowledege, and shows that there is a lot more to it then simply adding a search engine result. It is shown that if a human types in one query, a model can do the same but faster and with more instant diversification of the search query. 1 shows effectively how well even the smallest model of OpenAi is able to crank up the diversity of search queries. Later in 2 it can be seen clearly that the regular approach of "simple" adding a search engine result to the context is outperformed by the more sophisticated approach of augmenting the queries / diversifying the queries. Although there is still one question where the regular approach wins, it can be said that a more diverse input of website links, highlighting different aspects of the topic will lead to more diverse summary. Which is mirrored in the given ratings by the evaluator.

**The Evaluator** It is clear that such an evaluation as done here in this work brings up points to discuss, whether such an evaluation should be done, can be done or is even a scientific method which is allowed to use. The resarch on this topic is just starting and there are multiple compnies and research groups working on this. [Cha+23; Had+24; Kim+24]

**Time and efficiency** This is not the best solution for every query, just for very complex questions asked by the user. Simple queries should not be answered by such a solution which needs as many resources as this does. The time and energy to find the answer on "What is the capital of Italy?" searched in a Knowledge Graph is a fraction of what this system needs. For comparison a Knowledge Graph like Wikidata would need under .5 second, the regular search with summary took the system 9.28 seconds and the augmented approach took the system 17.95 seconds.

### 5.1 Limitations of Solution

The solution is limited by the price and time you want to wait for a solution. Another limitation is reproducibility, there are ways but in the end the language model can produce non deterministic outputs due to the settings of the model. The LLM evaluation is highly dependent on the model, the settings, the setup of the prompt template, how the questions are asked, how the answers are presented to the model, if the model needs to reason before answering, how the wording of the parameter which the models has to rate are, and many more. This work is limitations are due to the up-to-dateness of the topic, and the not up-to-dateness of the research surrounding this topic. The solution is not suited for easy queries, medium and very simple knowledge questions. The time is a limiting factor as well as access to the model and ability to execute a python script.

### 5.2 Lesson Learned - Conclusion

I learned that setting up a simple search in python with duckduckgo API is easy, setting up the model and suammrization is also a task which is fairly easily achieved. The more complex and tricky parts are the design choices when programming such a solution to ensure transparency. Showing the user what is done, and what the steps are the solution is taking was challenging. Evaluation was challenging as well, finding an approach which does not put me into place labeling all 20 summarizations results over and over again. I think that there is a lot of potential in such an evaluation appraoch, but there needs to be much more research done before using it as a reliable method. There needs to be more transparency to the output of the model. Letting the model reason before rating the A and B solutions is one method, but it is by far not the only and it is not clear if this is sufficient.

The shown solution does consume a lot of resources, and it showcases what could happen behind the curtains of OpenAi and Google. They are investing a lot of money to provide good search results, but using an LLM is simply much more expensive then just using the google search engine. Generation of tokes is expensive, and knowledge graphs like Wikidata can answer a lot of questions quicker and more cost efficient. Therefore this is just one part of a hybrid solution to provide good search results.

The context added to the language model was limited to the "title", "body" and "URL" of the search engine result. This means that only the first 50-75 words which are contained in the "body" of the website, will be inserted into the context window of the llm. The results are therefore highly depended on this aggregated space. This includes the risk of compressing complex questions into short terms and offers the risk of providing incomplete information. It should be considered using a crawler to grasp the full content of a website instead of taking the short snapshot of the website using the "body", "title" and "URL" for context.

In terms of time, the generation of tokens is the process which takes the longest, and it makes no difference whether a question is complex or trivial, it is more

about the length of the generated answer. Therefore this system is outperforming the human ability of search for complex topic at least by the time parameter.

**Conclusion:** The provided solution demonstrated that complex questions are answered more effectively when the LLM conducts multiple searches independently, compared to using the initial user query. For less complex questions, the performance difference becomes marginal. To answer the initial question: "Are LLMs better at using search engines than humans?" - it depends. It was shown that a simple "Yes" would not hold true for very simple questions. The human typed query would be faster and still would get the correct answer. When it comes to complex questions the shown approach of augmenting and diversifying the query, clearly outperforms the human, shown in 1. It is recommended to use such sophisticated systems when tackling hard problems and very complex situations. Even then it can be criticised that the resource usage is too high and that the system can still misrepresented results.

### 5.3 Future Work

Future work should look further into the evaluation of LLMs, evaluation of LLMs that evaluate and what the sweet spot is for augmenting the context window of LLMs and how the efficiency of knowledge graphs can be utilized to ground language models while being more cost efficient / energy efficient.

### 5.4 Github Code

For more details and the full code, please visit our GitHub repository: `https://github.com/nilskluewer/LLM-vs-Human-Search-Performance`.

### 5.5 Contributions

Large Language models were used to do basic grammar checks and to write a concise summary of this report - which is seen in the abstract. Everything else is self written and can therefore contain some mistakes. I tried rewriting the abstract myself but failed multiple times to write an abstract as good as the model did. (GPT-4o was used through the Playground OpenAi API)

## 6 Bibliography

## References

[Beu+23] Luca Beurer-Kellner et al. *Prompt Sketching for Large Language Models*. Nov. 8, 2023. DOI: 10.48550/arXiv.2311.04954. arXiv: 2311.04954[cs]. URL: http://arxiv.org/abs/2311.04954 (visited on 05/16/2024).

[Cha+23] Chi-Min Chan et al. *ChatEval: Towards Better LLM-based Evaluators through Multi-Agent Debate*. Aug. 14, 2023. DOI: 10.48550/arXiv.2308.07201. arXiv: 2308.07201[cs]. URL: http://arxiv.org/abs/2308.07201 (visited on 05/23/2024).

[Liu+23] Yuxuan Liu et al. *Calibrating LLM-Based Evaluator*. Sept. 23, 2023. DOI: 10.48550/arXiv.2309.13308. arXiv: 2309.13308[cs]. URL: http://arxiv.org/abs/2309.13308 (visited on 05/16/2024).

[Ope23] OpenAi. *ChatGPT - Homepage*. Nov. 18, 2023. URL: https://chat.openai.com (visited on 11/18/2023).

[XFC23] Ruiyun Xu, Yue Feng, and Hailiang Chen. *ChatGPT vs. Google: A Comparative Study of Search Performance and User Experience*. July 3, 2023. DOI: 10.48550/arXiv.2307.01135. arXiv: 2307.01135[cs]. URL: http://arxiv.org/abs/2307.01135 (visited on 04/29/2024).

[Had+24] Rishav Hada et al. *Are Large Language Model-based Evaluators the Solution to Scaling Up Multilingual Evaluation?* Feb. 13, 2024. DOI: 10.48550/arXiv.2309.07462. arXiv: 2309.07462[cs]. URL: http://arxiv.org/abs/2309.07462 (visited on 05/23/2024).

[Kim+24] Seungone Kim et al. *Prometheus 2: An Open Source Language Model Specialized in Evaluating Other Language Models*. May 2, 2024. DOI: 10.48550/arXiv.2405.01535. arXiv: 2405.01535[cs]. URL: http://arxiv.org/abs/2405.01535 (visited on 05/23/2024).

[Teo24] Sheila Teo. *How I Won Singapore's GPT-4 Prompt Engineering Competition*. Towards Data Science. Apr. 21, 2024. URL: https://towardsdatascience.com/how-i-won-singapores-gpt-4-prompt-engineering-competition-34c195a93d41 (visited on 05/16/2024).

[LMQ] LMQL LMQL. *LMQL is a programming language for LLM interaction. | LMQL*. URL: https://lmql.ai/ (visited on 05/16/2024).

[Ope] OpenAi. *OpenAI Platform - Prompt engineering*. URL: https://platform.openai.com (visited on 05/16/2024).

## 7 Appendix

```
tools = [
    {
        "type": "function",
        "function": {
            "name": "duckduckgo_text_search",
            "description": "Queries Duckduckgo search engine
    and responses with the top pages showing up!",
            "parameters": {
                "type": "object",
                "properties": {
                    "keywords": {
                        "type": "string",
```

```
                        "description": """Shortend - pls
    lookup in Codebase""",
                    },
                    "region": {
                        "type": "string",
                        "description": """Shortend - pls
    lookup in Codebase"""
                    },
                },
                "required": ["keywords"],
            },
        },
    }
]
"""
```

**Listing 1.6.** tools object which is passed to the model to give instructions on the usage of search function

```
"""
Generate 10 queries which are based of this description of
    queries:

<description>
We therefore move from easy questions up to complex questions.
     Complexity is in this context refereed to a question
    which will not yield in a high Precision due to the
    sponginess of the query, like : "What is the meaning of
    live".
</description>

We use this queires for evaluation of two systems which both
    can peform a search engine request. Move from easy to more
     advanced queries in terms of defined complexity.
"""
```

**Listing 1.7.** Prompt to generate questions for the evaluation

```
   system_message = f"""
        <query>
        {query}
        </query>

        <result_A>
        {json.dumps(enhanced_results, indent=2)}
        </result_A>

        <result_B>
        {json.dumps(regular_results, indent=2)}
        </result_B>

```

```
    First, reason out the differences between both results and why one could be better, or if they are both good. This should be in the "REASONING" field. Then, evaluate each set of results based on the following criteria:
        - Coherence: Logical consistency and fluency of text. How well does the results maintains a coherent and meaningful response. Give a rating from 1-10.
        - Relevance: How well the responses address the query specific information need. Give a rating from 1-10.
        - Completeness: Does the response cover all the essential information required to adequately answer the query? Give a rating from 1-10


    Use the template below to provide your evaluation. Not using this template will result in an ERROR. Please use the template strictly. Do not include
    the XML Tags.

    {
        "REASONING": "text with reasoning",
        "A": {
            "Coherence": <rating>",
            "Relevance": <rating>",
            "Completeness": <rating>"

        },
        "B": {
            "Coherence": <rating>",
            "Relevance": <rating>",
            "Completeness": <rating>"
        }
    }
    """
```

**Listing 1.8.** Prompt to generate requests for the evaluation

| query | reasoning | A_Coherence (AUGMENTED) | A_Relevance (AUGMENTED) | A_Completeness (AUGMENTED) | B_Coherence (REGULAR) | B_Relevance (REGULAR) | B_Completeness (REGULAR) | |
|---|---|---|---|---|---|---|---|---|
| What is the capital of Italy? | Both results provide a clear a | 9 | 9 | 10 | 8 | 9 | 9 | |
| What is the capital of Italy? | Both results provide a clear a | 8 | 8 | 9 | 9 | 9 | 9 | |
| What is the capital of Italy? | Both results provide a clear a | 9 | 9 | 10 | 8 | 9 | 8 | |
| What is the capital of Italy? | Both results provide a clear a | 9 | 9 | 9 | 8 | 9 | 8 | |
| What is the capital of Italy? | Both results provide a clear a | 9 | 9 | 9 | 8 | 8 | 8 | |
| | | 8.8 | 8.8 | 9.4 | 8.2 | 8.8 | 8.4 | MEAN |
| | | 0.6 | 0 | 1 | -0.6 | 0 | -1 | MEAN DIFF |
| Best laptops for students 2023 | Both results provide a list of | 9 | 9 | 9 | 8 | 7 | 7 | |
| Best laptops for students 2023 | Both results provide a list of | 8 | 9 | 8 | 9 | 8 | 7 | |
| Best laptops for students 2023 | Both results provide a list of | 9 | 9 | 9 | 7 | 7 | 7 | |
| Best laptops for students 2023 | Both results provide a list of | 8 | 9 | 8 | 9 | 8 | 9 | |
| Best laptops for students 2023 | Both results provide a list of | 7 | 7 | 8 | 8 | 9 | 8 | |
| | | 8.2 | 8.6 | 8.4 | 8.2 | 7.8 | 7.6 | MEAN |
| | | 0 | 0.8 | 0.8 | 0 | -0.8 | -0.8 | MEAN DIFF |
| Details of the Treaty of Versailles | Both results provide a list of | 9 | 9 | 9 | 8 | 8 | 8 | |
| Details of the Treaty of Versailles | Both results provide a list of | 9 | 9 | 9 | 8 | 8 | 8 | |
| Details of the Treaty of Versailles | Both results provide a list of | 9 | 9 | 9 | 8 | 8 | 7 | |
| Details of the Treaty of Versailles | Both results provide a list of | 9 | 9 | 9 | 8 | 8 | 8 | |
| Details of the Treaty of Versailles | Both results provide a list of | 9 | 9 | 9 | 8 | 8 | 8 | |
| | | 9 | 9 | 9 | 8 | 8 | 7.8 | MEAN |
| | | 1 | 1 | 1.2 | -1 | -1 | -1.2 | MEAN DIFF |
| Results of the 2022 FIFA World Cup final | Both results provide detailed | 9 | 9 | 9 | 8 | 9 | 8 | |
| Results of the 2022 FIFA World Cup final | Both results provide detailed | 9 | 9 | 9 | 8 | 8 | 8 | |
| Results of the 2022 FIFA World Cup final | Both results provide detailed | 9 | 9 | 9 | 8 | 9 | 8 | |
| Results of the 2022 FIFA World Cup final | Both results provide detailed | 9 | 9 | 9 | 8 | 9 | 8 | |
| Results of the 2022 FIFA World Cup final | Both results provide detailed | 9 | 9 | 9 | 8 | 8 | 8 | |
| | | 9 | 9 | 9 | 8 | 8.6 | 8 | MEAN |
| | | 1 | 0.4 | 1 | -1 | -0.4 | -1 | MEAN DIFF |
| How does blockchain technology work? | Both results provide a compr | 9 | 8 | 7 | 7 | 9 | 9 | |
| How does blockchain technology work? | Both results provide a detaile | 8 | 8 | 7 | 9 | 9 | 9 | |
| How does blockchain technology work? | Both results aim to explain h | 8 | 8 | 7 | 9 | 9 | 9 | |
| How does blockchain technology work? | Both results provide a compr | 8 | 9 | 8 | 9 | 9 | 9 | |
| How does blockchain technology work? | Both results provide a compr | 8 | 8 | 7 | 9 | 9 | 9 | |
| | | 8.2 | 8.2 | 7.2 | 8.6 | 9 | 9 | MEAN |
| | | -0.4 | -0.8 | -1.8 | 0.4 | 0.8 | 1.8 | MEAN DIFF |
| What are the pros and cons of keto diet | Both results provide a compr | 9 | 9 | 10 | 8 | 8 | 8 | |
| What are the pros and cons of keto diet | Both results provide a compr | 9 | 9 | 10 | 8 | 8 | 8 | |
| What are the pros and cons of keto diet | Both results provide a detaile | 9 | 9 | 10 | 7 | 8 | 8 | |
| What are the pros and cons of keto diet | Both results provide a compr | 9 | 9 | 10 | 8 | 8 | 8 | |
| What are the pros and cons of keto diet | Both results provide a detaile | 9 | 9 | 10 | 7 | 8 | 8 | |
| | | 9 | 9 | 10 | 7.6 | 8 | 8 | MEAN |
| | | 1.4 | 1 | 2 | -1.4 | -1 | -2 | MEAN DIFF |
| Impact of Brexit on UK economy | Both results provide a compr | 9 | 9 | 9 | 7 | 8 | 7 | |
| Impact of Brexit on UK economy | Both results provide valuable | 9 | 9 | 10 | 8 | 8 | 7 | |
| Impact of Brexit on UK economy | Both results provide valuable | 9 | 9 | 9 | 7 | 8 | 7 | |
| Impact of Brexit on UK economy | Both results provide valuable | 9 | 9 | 9 | 7 | 8 | 7 | |
| Impact of Brexit on UK economy | Both results provide valuable | 9 | 9 | 9 | 7 | 8 | 7 | |
| | | 9 | 9 | 9.2 | 7.2 | 8 | 7 | MEAN |
| | | 1.8 | 1 | 2.2 | -1.8 | -1 | -2.2 | MEAN DIFF |
| Which is better for home use: solar pane | Both results aim to address t | 9 | 9 | 10 | 7 | 7 | 6 | |
| Which is better for home use: solar pane | Both results aim to address t | 9 | 9 | 10 | 8 | 8 | 7 | |
| Which is better for home use: solar pane | Both results provide a compa | 9 | 9 | 10 | 8 | 7 | 6 | |
| Which is better for home use: solar pane | Both results aim to address t | 9 | 9 | 10 | 7 | 7 | 6 | |
| Which is better for home use: solar pane | Both results aim to address t | 9 | 9 | 10 | 8 | 8 | 7 | |
| | | 9 | 9 | 10 | 7.6 | 7.4 | 6.4 | MEAN |
| | | 1.4 | 1.6 | 3.6 | -1.4 | -1.6 | -3.6 | MEAN DIFF |
| Is artificial intelligence beneficial to soci | Both results provide a compr | 9 | 9 | 10 | 7 | 8 | 7 | |
| Is artificial intelligence beneficial to soci | Both results provide a compr | 9 | 9 | 10 | 7 | 8 | 7 | |
| Is artificial intelligence beneficial to soci | Both results provide a compr | 9 | 9 | 9 | 7 | 8 | 7 | |
| Is artificial intelligence beneficial to soci | Both results provide a compr | 9 | 9 | 10 | 7 | 8 | 7 | |
| Is artificial intelligence beneficial to soci | Both results provide a compr | 9 | 9 | 10 | 7 | 8 | 7 | |
| | | 9 | 9 | 9.8 | 7 | 8 | 7 | MEAN |
| | | 2 | 1 | 2.8 | -2 | -1 | -2.8 | MEAN DIFF |
| What is the best way to achieve a work- | Both results provide compre | 9 | 9 | 9 | 8 | 8 | 8 | |
| What is the best way to achieve a work- | Both results provide compre | 9 | 9 | 9 | 8 | 8 | 8 | |
| What is the best way to achieve a work- | Both results provide compre | 9 | 9 | 9 | 7 | 8 | 8 | |
| What is the best way to achieve a work- | Both results provide compre | 9 | 9 | 9 | 8 | 8 | 8 | |
| What is the best way to achieve a work- | Both results provide compre | 9 | 9 | 9 | 7 | 8 | 8 | |
| | | 9 | 9 | 9 | 7.6 | 8 | 8 | MEAN |
| | | 1.4 | 1 | 1 | -1.4 | -1 | -1 | MEAN DIFF |
| | | | | | | | | |
| | | 8.6 | 5.2 | 10.8 | -8.6 | -5.2 | -10.8 | OVERALL DIFF |

**Fig. 2.** Evaluation Results