GPT-ReXplorer: Recursive Topic Expansion & Web Crawling Explorer for GPT Model Training

Benjamin Rader
IUPUI
Purdue School of Engineering & Technology

Abstract—In this paper, I present a novel approach to automating the training process of GPT models on various internet data sources, with the aim of enhancing problem-solving capabilities during chat-based interactions. The proposed ReXplorer, a Python-based local program, utilizes topic extraction and expansion methods to streamline GPT model training. Although the initial implementation of ReXplorer did not yield the desired results, it provided valuable insights into the challenges associated with information retrieval and the inherent limitations of search engines. These findings highlight the potential for future innovations in creating intelligent training systems for GPT models, ultimately transforming how we solve problems.

1 Introduction & Literature Overview

The world is about to change a lot. The AI race is arguably already afoot, and it is unclear where it will go. This year will mark a change in problem-solving approaches, the way developers debug their code, and soon enough the way we obtain information. The internet is a collection of devices communicating with each other enabling us to approach various goals as humanity. Although our goals may differ, societies generally have collections of people who share the same goals and morals. The internet is a means to reach such goals. It allows us to share knowledge, communicate, store information, compute, and solve problems. The internet has been around for quite a while now and many would say that it has connected people in ways unimaginable. However, we can flip that script and look at it a bit more critically and ask if we really are more connected. I would argue that to be connected means we are aware of reality and have the ability to traverse it somewhat. In a practical view, this simply means doing things like reading books, talking to people, observing the world with your senses, or even sitting and contemplating reality. This is how we gain knowledge, how we solve problems, and how we attempt to achieve our goals. Once again, the internet is a medium in which we can do such things. However, is it really that great at helping us traverse reality? A philosophical way to phrase this would be ”ontology traversal.” In this paper, I will hypothesize a view of the purpose of the internet, the means of traversing ontology or reality with the internet's use, show how AI is the natural next step to doing so and talk of the shortcomings and limitations of current GPT models from OpenAI. Lastly, I put forth a naive approach to temporarily solve the issue with a Python-based web crawling system. The implementation piece is not polished enough to solve the complete issue, but it is a proof of concept for how individuals could approach future problem-solving with the help of GPT models.

1.1 Philosophy of the Internet

In order to approach the issue that is to be talked about in this paper, we must start broadly with the purpose and the internet and then delve deeper into the purpose and the methods for fulfilling such purposes. There are many philosophers who talk about Teleology. Teleology is concerned with the perceived ends or goals of actions, events, processes, or really anything, and how they are related to their causes or their means [20]. When discussing the teleology of the internet, it is easy to say it is meant for this or for that, but regardless ethics suggests that we ought to use the internet in ways that help us achieve our goals. In the most general sense, the internet is meant to fulfill our goals, and it is as simple as that. Many agree that this goal is to connect people, entertain people, and compute things for people. However, these conclusions are somewhat shallow and do not fully describe one of the main goals of the internet. To translate all of these purposes, the internet has the most basic purpose of changing how we view reality. The end or goal is to change our understanding or perception of reality. This would also relate to ontology which is, more or less, the study of the nature of reality.

When people view a picture or read something on the internet, they change their view of reality. Obtaining knowledge consequentially seems to be the main purpose of the internet. However, to say that a person obtains knowledge may not make sense. Our brains are more like RAM drives than they are piggy banks. It would be more analogous to say that the internet refactors our knowledge configurations. In order to constantly change our knowledge configurations, we must ingest new content with our senses as it is obtained through the internet. Peoples' individual motivations will differ for why they are on the internet too, but ultimately it is to fulfill goals or interests of the individual.

We use to internet to learn, communicate, and change our perceptions of reality. However, the ”knowledge configuration” that results is sometimes not what is needed to fulfill our goals. For instance, a programmer may need to write code that can run on particular hardware, but they cannot seem to find the answer to the problem. The first thing the programmer will do it go onto the internet. Subsequently, they will use ”information retrieval” to retrieve the information they need to fulfill their goal.

1.2 Information Retrieval

By definition, information retrieval is usually ”a software program that deals with the organization, storage, retrieval, and evaluation of information from document repositories, particularly textual information. Information Retrieval is the activity of obtaining material that can usually be documented on an unstructured nature [2]” Most of the information that will be used for problem-solving or learning on the internet involves unstructured textual data. Even Youtube videos have titles and unstructured metadata. Eventually, information retrieval is how we navigate ontologies (knowledge configurations) on the internet.

1.3 Ontology Traversal and AI

1.3.1 The Ontology Landscape Allegory

Before a person does multiplication, they generally have to understand addition or subtraction. When a person navigates into a place where they can understand certain abstractions, this could be called a particular ontology or view of reality. Information retrieval allows us to use the internet efficiently to traverse these ontologies.

I now hypothesize a sort of allegory that can be used to help solve problems of information retrieval on the internet and how to approach them. This allegory was the motivation for this paper.

Imagine that reality or ontology is a landscape with hills, rivers, and all the makings of a natural world. Traversing this landscape is synonymous with ontology traversal, changing one’s knowledge configuration, or even reading an article online. The real issue comes with how we traverse the landscape. It is difficult to reach the peak of a mountain that could represent understanding an abstraction such as that of high-level calculus. Doing so without transportation or help would be difficult, if not impossible. Luckily, we have bridge builders and construction workers - other people. Let’s say that an island represents someone’s knowledge and they need to get to another island to understand something. To find those islands they have to use the search engine. However, if the island doesn’t have a direct bridge to it, then the person may have to travel across multiple islands (ontologies or abstractions) till they arrive at that place. Search engines are what we use to find out which ”bridges” are touching our ”islands.” The bridges are webpages, articles, videos, and things that we can find on the internet with those search engines.

1.3.2 Search Engines & Information Retrieval

Search engines allow us to find the information that we need so long as we use the correct terminology in our queries. This is why multiple ”islands” sometimes need to be traversed to arrive at our destination. Multiple StackOverflow posts or YouTube videos may be necessary to bridge our knowledge gap before we can solve a coding problem. However, this approach is extremely inefficient. We are relying on the knowledge of certain terms. When certain areas of problem-solving involve high-level abstractions, it then becomes time-consuming to learn enough to understand the concepts deeply. This is depicted in Figure 1.

1.3.3 GPT & Information Retrieval

GPTs (generative pretrained transformers) are to search engines what tractors were to hand plows. Just as tractors revolutionized farming by making it faster, more efficient, and more productive, GPTs are revolutionizing information retrieval and epistemology (the study of knowledge and how we come to know things) by making it faster, more efficient, and more productive. GPT simply takes an input and is trained to give creative and long outputs [5], [11], [19]. I would hypothesize that GPT is only the beginning of the information retrieval revolution. GPT automates the ”bridge building.” If a person wants to learn something, then a large GPT model can bridge the gap of knowledge between the user’s understanding or context and what they need to know. Teachers have a uniquely difficult job of bridging the knowledge gaps of multiple students at once with only one curriculum or bridge. In other words, there will always be misunderstandings unless the students ask questions to get the information in the formation or abstraction that they need [9]. This representation of classroom learning is shown in Figure 2.

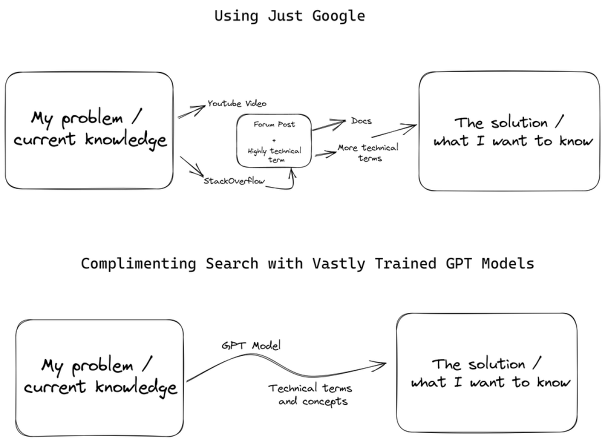

Using Just Google
My problem / current knowledge
Youtube Video
Forum Post + Highly technical term
StackOverflow
Docs
More technical terms
The solution / what I want to know
Complimenting Search with Vastly Trained GPT Models
My problem / current knowledge
GPT Model
Technical terms and concepts
The solution / what I want to know

Fig. 1. This represents the problem of ontology traversal when solving problems. A person may go through multiple sources before understanding a concept well enough.

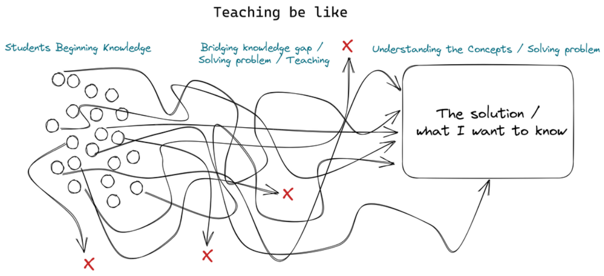

Teaching be like
Students Beginning Knowledge
Bridging knowledge gap / Solving problem / Teaching
Understanding the Concepts / Solving problem
The solution / what I want to know

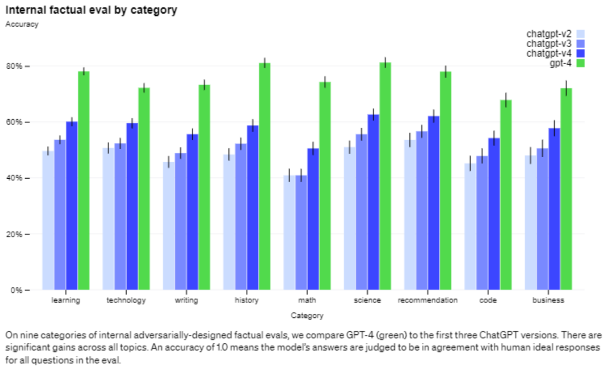
Fig. 2. Students must arrive at the teacher’s desired solution or level of understanding with only one curriculum. Students can still recorrect or get there if they ask questions, but most do not.

1.4 GPT Limitations and Misunderstandings

Going deep into the workings of OpenAI's GPT models, the datasets they used to train or evaluate them, and the nuances of using the API or out of the scope of this research. However, there is anecdotal evidence and a bit of published stats from OpenAI that were the primary motivation for this paper.

I have been using ChatGPT and the OpenAI API extensively since its inception. OpenAI has written about the limitations of its model when it comes to novel and abstract concepts even in the documentation for each model. More recently, they opened the use of GPT-4 to the public for their online chat interface. GPT-4 has orders more parameters than GPT-3 which amounts to 170 trillion total parameters. This number is hard to fathom. However, my experience with both GPT-3.5 and 4 have shown that there are limits.

GPT models do not "understand" or have knowledge in any way. In an abstract way, they take an input and predict the output. In my own experience, this quickly became apparent. When solving novel problems or building new software, GPT-4 even was not that great after the abstractions became too large or there are too many moving parts. These models can take in a lot of text, but when the concept becomes too abstract, it will start to "hallucinate." Even short prompts with non-abstract workings can cause these hallucinations. Data from OpenAI (Figure 3) exhibits these hallucinations with different models and topics to show that it is factually inaccurate or wrong a fifth (20%) of the time [19]. This means that GPT is not great for a lot of things: being used as a programmatic middleware in high-risk applications, medical applications, or spaces where the abstractions are a bit too large.



On nine categories of internal adversarially-designed factual evals, we compare GPT-4 (green) to the first three ChatGPT versions. There are significant gains across all topics. An accuracy of 1.0 means the model's answers are judged to be in agreement with human ideal responses for all questions in the eval.

Fig. 3. GPT model comparison of hallucinations where the model is wrong about a particular question (organized by fields)

1.5 The Problem and Motivation

I had heard many developers and people saying that GPT was not good enough to help them with something or that it was not good enough. I would argue that this is far from the case. Despite GPT having a sort of limit or "window" on its understanding of long abstract inputs, it seems that this is more of an issue of breaking down the problem analytically. For instance, GPT may not be able to tell you how to build a rocket, but GPT could help with the associated sub-problems of building a rocket. Therefore, it may be more productive to decrease hallucinations with training on specific problems, then use the GPT models in a more modular way when problem-solving. In the case of building a rocket, this could mean showing the GPT models papers, diagrams, and blueprints of rocket technology, then asking it specific questions related to that data.

I hypothesize that it would be effective and productive to utilize the technology at hand to automate the process of training GPT models on various internet data in the pursuit of solving modularized and specific problems during chatting.

1.6 Topic Extraction and Expansion Methods

To train GPT models, it would be best if we could simply tell it what to learn and have it crawl the internet to do so. To do so, there are two important methods that must be used: topic extraction and topic expansion. Topic extraction is a well-known method. However, topic expansion is not a name that is used by anyone. Rather, for topic expansion, there are GPT models for generating topics based on the information and then techniques such as "query expansion" which are used with search engines.

1.6.1 Topic Extraction

Topic extraction is taking key topics out of a corpus of text. This process can be done through various methods including clustering, classification, or NLP (natural language processing). Most of the new techniques for doing so with GPT involve smaller inputs though. For instance, there is one implementation that involved using GPT-3 and BERT (Bidirectional Encoder Representations from Transformers) to figure out the categories of items that people needed when shopping in an online store [15]. However, some methods could also take inputs along with a larger corpus of text. These techniques such as GuidedLDA (Latent Dirichlet Allocation) seem to be promising in that you can incorporate some guidance into what sorts of topics should be extracted [24]. To explain, LDA is a topic-modeling technique that uses unsupervised learning to identify topics based on the word distributions in the text. These methods work really well with long texts but poorly with short ones.

1.6.2 Topic Expansion

The expansion of topics is not a new idea, but it depends on what one means by topics. In this case, we can say that topics can involve sets of keywords and prompts or queries. In order to train a GPT model with new material and make it more innovative or creative, we have to use methods to develop new search queries. GPT models are really good for query expansion. One method used GPT-2 to generate text from a "seed query" and then uses this as input into a BM25 search system [10]. There is a lot of work that has been done in traditional methods too such as using ontologies (in the knowledge engineering sense), association rules, wordnet, a meta thesaurus, synonym mapping, local occurrence, and latent semantic indexing (LSI) [6]. There are even approaches that utilize knowledge bases and feedback during operation [7], [12], [22].

2 REXPLORER DESIGN AND IMPLEMENTATION

I designed a Python-based local program that can recursively web crawl and expand based on an initial set of parameters and guidance from the user.

2.1 Components and Architecture

Below I've listed the various components of the ReXplorer stack:
Superjob:

- A ”superjob” is centered around one set of inital inputs (URLs, prompts, topics)
- Superjobs keep track of the ”depth” (how many jobs have been run) and parameters which can stop the program if a limit is reached
- Superjobs run once and are not recursive

Job:

- Jobs can involve multiple URLs, a set a prompts, or distributions of keywords (topics)
- Expansion is done at the level of collections of URLs and their data, so expansion is dependent upon all of the data at each depth
- Jobs start with a current set of URLs, prompts, and topics, but they also take in the initial set of data to keep on track and not expand too much

Data Acquisition (Scraping):

- Uses ”scrapy” to scrape lists of URLs

Topic Extraction:

- Uses LDA and Guided LDA [23] along with named entity resolution (NER)

Topic Expansion:

- Uses text-davinci-003 (GPT-3) to generate new search queries based on current prompts, topics, and URLs.
- During generation it also takes the previous set of prompts and the initial prompts into account

Folder Structure:

- The hardest part to design. Every part of the program refers to certain filenames and UUIDs.
- Most of the program utilize metadata files that are created during the process

The structure looks like

```
superjobs folder/
└─ superjob__<superjob_id>
   ├─ <job_num>__<depth>__<superjob_id>
   │  ├─ data/
   │  │  ├─ example1.html
   │  │  └─ example2.html
   │  └─ ...
   └─ ...
```

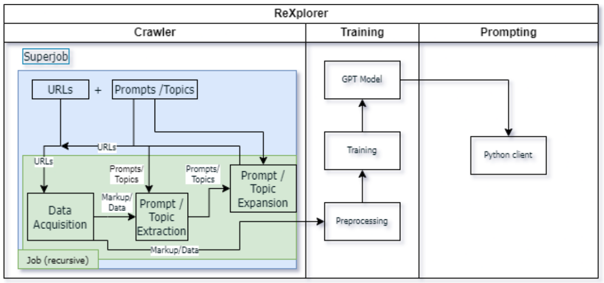


Fig. 4. Rexplorer Architecture: blue area - superjob (main loop), green (job loop)

2.2 Optimization and Safety Rails

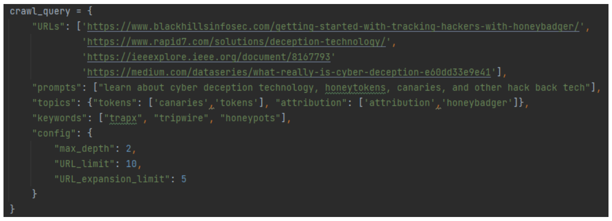
In order to keep training on track, it is desirable to constantly remind the GPT-3 model (during topic expansion) what the original topic is. Therefore, every time topic expansion occurs, we feed the GPT-3 model the initial prompt and keywords. Additionally, we need the prompts from the previous ”depth.” This encourages the GPT-3 to be creative with its query generation while still having a goal in mind.

For additional control, there were two parameters added to the expansion process that limit it purposefully: ”url_expansion_limit” and ”search_query_limit”. These enable the topic expansion module to prioritize by: 1) not outputting too many search queries and tokens from GPT-3 and 2) not generating too many URLs. However, it is important to note that the search queries are used to gather URLs via a search engine API. If the ”url_expansion_limit” parameter is 10 and the ”search_query_limit” is 10, then only the first result will be taken from each search since we are only allowed 10 outputted URLs, but must use 10 search queries. This sort of design also allows for flexibility based on the application. It allows the user to be more conservative on the search engine returns. Use 10 for the URL and 1 for the search query param means we will get 10 results from the search query. Ideally, we would implement more granularity here, but this is a start.

3 REXPLORER FINDINGS AND EVALUATION

The current ReXplorer has much to be desired. Even with the safety rails and a good bit of work into the prompt engineering for the topic expansion, the results indicate a clear need for further investigation. Below are the results and some of the issues.

3.0.1 Run of Rexplorer with 4 Security Pages



```
crawl_query = {
    "URLs": ['https://www.blackhillsinfosec.com/getting-started-with-tracking-hackers-with-honeybadger/',
             'https://www.rapid7.com/solutions/deception-technology/',
             'https://ieeexplore.ieee.org/document/8167793'
             'https://medium.com/dataseries/what-really-is-cyber-deception-e60dd33e9e41'],
    "prompts": ["learn about cyber deception technology, honeytokens, canaries, and other hack back tech"],
    "topics": {"tokens": ['canaries','tokens'], "attribution": ['attribution','honeybadger']},
    "keywords": ["trapx", "tripwire", "honeypots"],
    "config": {
        "max_depth": 2,
        "URL_limit": 10,
        "URL_expansion_limit": 5
    }
}
```

Fig. 5. **crawl_query** object that is used to initialize the super job with all of its parameters (URLs, prompts, topics, keywords, and config variables for the run.

This run used the above configuration to begin the recursive exploration. I understand, quite well, the realm of cyber deception, what it entails, and some of the good resources for the subject. However, it did not take much investigation to see that ReXplorer has some deficiencies. Below, in listing 1, you'll find some of the extracted keywords from some of the security articles. It's quite apparent that work needs to be done in the topic extraction part of the program to obtain more meaningful keywords. You can see that **security** and **wireless** are in it, but almost all of the other words except technically **adhd** (Active Defense Harbinger Distribution). These keywords don't seem to be useful for the most part so maybe this particular use of GuidedLDA/LDA was unnecessary or unpolished. The main reason for their use was to try pulling out terminology that could help search query creation, but most of the terms seem to be a bit useless.

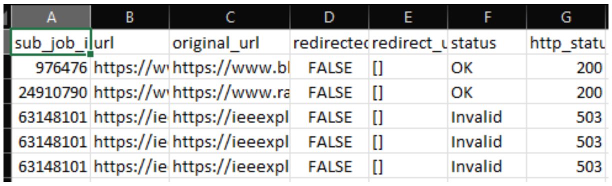
```
"topics": {
    "topic1": [
        "wireless",
        "actually",
        "security",
        "google",
        "give",
        "whenever",
        "mono",
        "document",
        "look",
        "thing"
    ],
    "topic2": [
        "see",
        "black",
        "adhd",
        "would",
        "time",
        "let",
        "team",
        "work",
        "map",
        "run"
    ],
```

Listing 1: Topics extracted from one of the cybersecurity pages

It seems that the extracted terms were also likely used disproportionately to affect GPT-3's output, which is why we also get non-meaningful URLs as a result like dictionary results. However, when thinking about search engine results, dictionary results are almost always at the top of the list. This brings up issues that I had not expected or would have never expected. It seems that training a GPT model on such data would be to no avail in getting innovative findings and problem-solving opportunities.

3.0.2 *Potential Improvements*

1) Improve Web Scraping
   - Web scraping is a game of cat and mouse, so it comes to no surprise that the research paper URLs gave errors. This would require a lot of focus, but improving this could mean for better future results with the data coming from data-rich sources such as research papers.
   - Scrapy was used for this program which has lots of custom middlewares that can help scrape websites that normally block attempts
2) Prompt Engineering
   - Finding a balance between a low amount of tokens/words, fitting within the token limit, and having accurate and desired output is difficult. Engineering a better prompt and more work in trying out different structures with the prompt could help save money and tokens while generating better search queries
   - Giving it more programmatic and conditional reactions to the input data could help generate better search queries. For instance, telling it to ignore certain topic words if they seem unrelated to the initial prompt.
3) Search Engine Nuances
   - Search engines like Google which was used in this experiment have a vast number of things to account for. This is why, I argue, search engines are a bit outdated. The monetization model for them and how they return results are sub-par for this use case.
   - Account for the tendencies of certain topics to bring back dictionary results that are not productive with training
   - Use layers of different GPT models to help formulate better queries that can anticipate and react to some of the search engine nuances.



A	B	C	D	E	F	G
sub_job_i	url	original_url	redirectec	redirect_u	status	http_statu
976476	https://w	https://www.bl	FALSE	[]	OK	200
24910790	https://w	https://www.ra	FALSE	[]	OK	200
63148101	https://ie	https://ieeexpl	FALSE	[]	Invalid	503
63148101	https://ie	https://ieeexpl	FALSE	[]	Invalid	503
63148101	https://ie	https://ieeexpl	FALSE	[]	Invalid	503

Fig. 6. **crawl_query** object that is used to initialize the super job with all of its parameters (URLs, prompts, topics, keywords, and config variables for the run.

4 Conclusion

Large language models are a good start to integrating AI into our daily lives in monumental ways. However, even the slightest improvements over some of their obvious limitations will take a lot of work and innovation. The dilemma of training AI is that we are limited by our ability to obtain meaningful information. With GPT-4 and the inevitable larger or future GPT-5 model, it will be able to answer very innovative questions, but we will one day need a system that can automatically learn about certain topics by web crawling. Unfortunately, the fact that most of this takes place with a search engine creates a lot of inherent issues, mostly because of models and systems that put certain results to the top. In conclusion, information retrieval is difficult, and although ReXplorer was merely an ”attempt” at trying to create a smart training system for GPT, it shows that more innovation in certain areas could lend to a system that changes how we solve problems.

REFERENCES

[1] Ethics of technological disruption.

[2] What is a knowledge graph?

[3] Knowledge engineering, 05 2019.

[4] Siddhi . What is information retrieval?, 07 2020.

[5] 262588213843476. Everything i understand about chatgpt, 02 2023.

[6] Lasmedi Afuan, Ahmad Ashari, and Yohanes Suyanto. A study: query expansion methods in information retrieval. *Journal of Physics: Conference Series*, 1367:012001, 11 2019.

[7] Hiteshwar Kumar Azad, Akshay Deepak, Chinmay Chakraborty, and Kumar Abhishek. Improving query expansion using pseudo-relevant web knowledge for information retrieval. *Pattern Recognition Letters*, 158:148–156, 06 2022.

[8] Jack Bandy. Dirty secrets of bookcorpus, a key dataset in machine learning, 05 2021.

[9] Christine Chin. Students' questions: fostering a culture of inquisitiveness in science classrooms. *The School science review*, 86:107–112, 2004.

[10] Vincent Claveau. Query expansion with artificially generated texts. *arXiv:2012.08787 [cs]*, 12 2020.

[11] Kindra Cooper. Openai gpt-3: Everything you need to know, 11 2021.

[12] Julio Hernandez, Heidy M. Marin-Castro, and Miguel Morales-Sandoval. A semantic focused web crawler based on a knowledge representation schema. *Applied Sciences*, 10:3837, 05 2020.

[13] Rohan Jagtap. Openai gpt: Generative pre-training for language understanding, 07 2020.

[14] Daniel Martin Katz, Michael James Bommarito, Shang Gao, and Pablo Arredondo. Gpt-4 passes the bar exam, 03 2023.

[15] Su Young Kim, Hyeonjin Park, Kyuyong Shin, and Kyung-Min Kim. Ask me what you need: Product retrieval using knowledge from gpt-3. *arXiv:2207.02516 [cs]*, 07 2022.

[16] Ryosuke Kinoshita and Shun Shiramatsu. Agent for recommending information relevant to web-based discussion by generating query terms using gpt-3, 11 2022.

[17] Anis Koubaa. Gpt-4 vs. gpt-3.5: A concise showdown. *Preprints.org*, 03 2023.

[18] Mohammed Lubbad. The ultimate guide to gpt-4 parameters: Everything you need to know about nlp's game-changer, 03 2023.

[19] OpenAI. Gpt-4, 03 2023.

[20] Robert Pasnau. Thomas aquinas, 2023.

[21] Jayr Pereira, Robson Fidalgo, Roberto Lotufo, and Rodrigo Nogueira. Visconde: Multi-document qa with gpt-3 and neural reranking. *arXiv:2212.09656 [cs]*, 12 2022.

[22] Dilip Kumar Sharma, Rajendra Pamula, and Durg Singh Chauhan. Query expansion – hybrid framework using fuzzy logic and prf. *Measurement*, 198:111300, 07 2022.

[23] Vikash Singh. Guidedlda: Guided topic modeling with latent dirichlet allocation, 04 2023.

[24] Manju Venugopalan and Deepa Gupta. An enhanced guided lda model augmented with bert based semantic strength for aspect term extraction in sentiment analysis. *Knowledge-Based Systems*, 246:108668, 2022.