



# HyperBrain: Human-inspired Hypermedia Guidance using a Large Language Model

Danai Vachtsevanou  
University of St. Gallen  
St. Gallen, Switzerland  
danai.vachtsevanou@unisg.ch

Jérémy Lemee  
University of St. Gallen  
St. Gallen, Switzerland  
jeremy.lemee@unisg.ch

Raffael Rot  
University of St. Gallen  
St. Gallen, Switzerland  
raffael.rot@student.unisg.ch

Simon Mayer  
University of St. Gallen  
St. Gallen, Switzerland  
simon.mayer@unisg.ch

Andrei Ciortea  
University of St. Gallen  
St. Gallen, Switzerland  
andrei.ciortea@unisg.ch

Ganesh Ramanathan  
University of St. Gallen  
St. Gallen, Switzerland  
ganesh.ramanathan@student.unisg.ch

## ABSTRACT

We present HyperBrain, a hypermedia client that autonomously navigates hypermedia environments to achieve user goals specified in natural language. To achieve this, the client makes use of a large language model to decide which of the available hypermedia controls should be used within a given application context. In a demonstrative scenario, we show the client's ability to autonomously select and follow simple hyperlinks towards a high-level goal, successfully traversing the hypermedia structure of Wikipedia given only the markup of the respective resources. We show that hypermedia navigation based on language models is effective, and propose that this should be considered as a step to create hypermedia environments that are used by autonomous clients alongside people.

## CCS CONCEPTS

• Information systems → World Wide Web; • Human-centered computing → Interactive systems and tools; • Computing methodologies → Natural language processing.

## KEYWORDS

Hypermedia, Large Language Model, HATEOAS, Web of Things

### ACM Reference Format:

Danai Vachtsevanou, Jérémy Lemee, Raffael Rot, Simon Mayer, Andrei Ciortea, and Ganesh Ramanathan. 2023. HyperBrain: Human-inspired Hypermedia Guidance using a Large Language Model. In *34th ACM Conference on Hypertext and Social Media (HT '23)*, September 4–8, 2023, Rome, Italy. ACM, New York, NY, USA, 5 pages. <https://doi.org/10.1145/3603163.3609077>

## 1 INTRODUCTION

In his article *The Computer for the 21st Century*, Mark Weiser opens with the statement that “The most profound technologies are those that disappear” [8]. Weiser conveys how truly impactful technologies seamlessly integrate into our lives to the extent that we no

longer consciously recognize them as technology at all. This concept forms the foundation of ubiquitous computing, which proposes that computing devices could be present anywhere in the environment and that humans could interact with these devices in a more natural fashion — famously, this might “make using a computer as refreshing as taking a walk in the woods.” [8].

Hypermedia systems can significantly contribute to achieving the vision of ubiquitous computing by providing users with intuitive, and flexible nonlinear access to an extensive range of information and services distributed throughout the environment. However, as hypermedia systems such as the Web grow in complexity (even extending to the physical world<sup>1</sup>), it becomes challenging for human users to navigate them efficiently without additional support. For this, tools and services such as search engines were developed to facilitate exploration by reducing the number of hyperlinks users need to navigate, ideally providing the most relevant hyperlink based on a user goal, often expressed as a textual query. Nowadays, search engines, such as Google, use Large Language Models (LLMs) to better understand user queries and increase the relevance of the search results [1]. Search engines significantly assist users in discovering a relevant entry point upon starting their navigation. Yet, such guidance focuses on providing initial search results and does not seamlessly integrate throughout the remaining navigation process towards users' goals.

Our guiding idea for this paper is that *an LLM might be used to provide hypermedia guidance*, that is, to indicate to a user which hypermedia control to use *during navigation* and *given the user's goal* when several controls are available. We argue that LLMs, with their capability to process text, such as user queries or hypertext, offer a promising solution to bridge the gap between users' goals and the run-time state of hypermedia environments, providing valuable guidance for goal-driven hypermedia navigation. We propose the use of LLMs in hypermedia systems for automating the matching of application goals with dynamically discovered hypermedia interfaces. This proposal draws inspiration from how human users navigate hypermedia environments, performing a similar process to align their objectives with available interfaces [5].

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).  
HT '24, September 4–8, 2023, Rome, Italy

© 2023 Copyright held by the owner/author(s).  
ACM ISBN 979-8-4007-0232-7/23/09.  
<https://doi.org/10.1145/3603163.3609077>

<sup>1</sup>The Web of Things initiative, driven by the World Wide Web Consortium, defines a set of standards for the integration of physical devices in the Web: <https://www.w3.org/WoT/>.

In the paper, following a short overview of background knowledge, we show that an LLM can effectively determine, when supplied with a set of hypermedia controls and the user's goal, *which hypermedia control is more likely to assist the user in pursuing their objective*. We begin exploring the potential of our approach by demonstrating LLM-based hypermedia guidance in a well-known environment, rich in plain hypertext: Wikipedia. Here, the user goal of finding a specific Wikipedia page is represented as a single term, and we initialize our system with another Wikipedia page; the system then uses an LLM to navigate hyperlinks that are provided by Wikipedia to ultimately find the target page.

## 2 BACKGROUND KNOWLEDGE

In the following, we offer an overview of how interaction takes place in hypermedia systems, identifying the key features required for LLM-based guidance that guides users from *local* hypermedia controls to *global* user goals<sup>2</sup>.

In hypermedia systems like the Web, users chain the usage of hypermedia controls, which can range from simple hyperlinks that are followed by clicking to more complex hypermedia forms that may require (sometimes validated) input. This chaining of interactions is rarely arbitrary or uninformed; instead, users typically *navigate* hypermedia with a clear sense of purpose, guided by the overarching direction — or *global guidance* — that is provided by their goal. For example, to buy a book online, a user first visits the Web page of a store where the book is sold, then adds the book to a (virtual) cart, possibly along with additional items, and finally proceeds to checkout. Inspired by how humans navigate hypermedia, we explore how LLMs can perform this goal-driven chaining of hypermedia controls. This involves interpreting the goal of a user and then predicting how the available hypermedia at an application state can help reach the state desired by the user, even in cases where multiple state transitions are needed in between.

Despite being driven by global user goals, interactions in hypermedia systems are also facilitated between state transitions. This more fine-grained *local guidance* is prevalent on the Web, where users encounter only a subset of hypermedia controls at a given application state, and this subset is updated after each interaction. For instance, when a user first visits the Web page of a bookstore, they may find interaction options to search for books or add books to the cart, while the option to checkout may not be available at that state. However, once the user interacts to add a book to the cart, a new set of hypermedia will be provided, including options to remove the book from the cart or proceed to checkout. Concretely, this local guidance is realized within the Web architecture through a principle that is part of the Uniform Interface constraint in the Representational State Transfer architectural style [2], namely “Hypermedia as the Engine of Application State” (HATEOAS). HATEOAS constrains the design of Web servers, where a successful request should result in a response containing a new set of hypermedia that indicates the available interactions in the updated application state. Web clients, in turn, should effectively interpret this hypermedia and make requests to the server by exploiting the hypermedia controls and creating suitable input values in the correct format.

Although HATEOAS effectively reduces the knowledge required on the client side to interact with servers, the induced local guidance is often *independent* of the user's specific goal or based solely on an *estimate* of the user's goal (e.g., in the case of Web services that link to videos or pictures that are recommended for a specific user). As a result, the user remains responsible for bridging the gap between HATEOAS-enabled local guidance and their global goal's guidance upon making their next interaction. Here, our aim is to leverage LLMs to, not only fulfill the responsibilities of typical Web clients, but also mimic human user behavior, resulting in an enhanced form of *local guidance that takes into consideration user goals*. For the former, LLM-based clients should interpret and exploit hypermedia controls based on content and formats delivered by servers (such as in HTML documents [3], OpenAPI specifications [7], or W3C Web of Things Thing Descriptions [6]). For the latter, they should compute the best link option based on the available hypermedia (as provided through HATEOAS) *and* the estimated relevance to a goal (as provided by a user). Finally, keeping a record of past interactions may aid in avoiding loops in navigation, and constructing more contextualized input for interactions (e.g. so that the checkout process includes only the ID of the product that the user is currently interested in buying).

## 3 HYPERMEDIA GUIDANCE WITH LARGE LANGUAGE MODELS

We propose to use LLMs for providing local guidance to machine clients in hypermedia environments. We argue that LLMs are well-suited to extract the implications of following a hypermedia control when given an appropriate context description (e.g., the label of a hyperlink in HTML).

### 3.1 HyperBrain: An LLM-Based Hypermedia Client

The main component of the system we present in this paper is HyperBrain<sup>3</sup> — an LLM-based hypermedia client that is capable of making decisions about how to navigate hypermedia environments to reach a specified user goal.

A session with HyperBrain is initialized with a *keyword* in a textual format that represents a user goal (e.g. “information about Rome”), and an *entry-point* URI that identifies the current application context, i.e. the application state that determines the current availability of hypermedia (e.g. a Wikipedia page). With these two parameters, HyperBrain starts its navigation through the hypermedia environment, aiming to transition from the initial application state to a state that satisfies the user's goal (e.g. the Wikipedia page about Rome). During navigation, HyperBrain can generate and submit queries to an LLM instance to support its decision-making process within specific application contexts. These queries serve the purpose of examining the hypermedia within the application context that is most likely related to the user's global goal, and determining whether the application context satisfies the goal.

We evaluate our approach using the Wikipedia environment, demonstrating how our LLM-based hypermedia client effectively

<sup>2</sup>We adopt the terms *local* and *global* guidance from [4] to explain how users' interactions are guided in hypermedia systems.

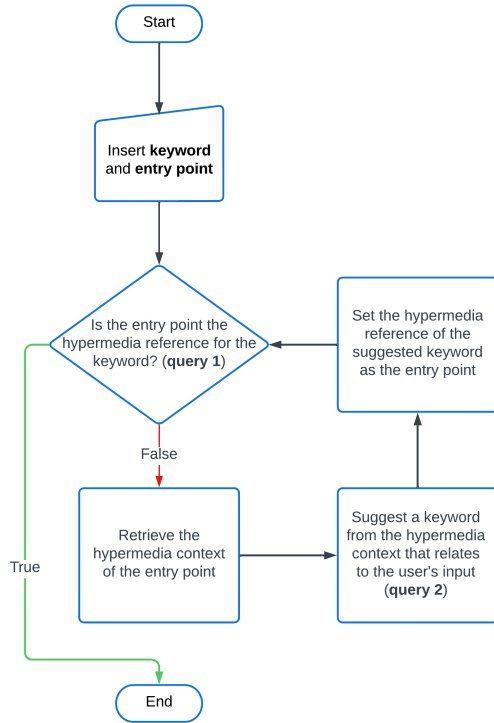
<sup>3</sup>HyperBrain is implemented in Python and its code is available online: <https://github.com/Interactions-HSG/hyperbrain>

```
query = f"Available link: '{entry_point}'" \
    f"Guess if the available link is the hypermedia reference " \
    f"for '{keyword}'." \
    f"Return with TRUE/FALSE."
```

**Code Listing 1: The client queries whether the current application state corresponds to the goal application state.**

```
query = f"List of keywords: '{context}'." \
    f"Guess which of these keywords is most likely related " \
    f"to '{keyword}'." \
    f"Provide a keyword and one-sentence explanation."
```

**Code Listing 2: The client queries which of the keywords in the current context is likely to help transition towards the user's goal.**



**Figure 1: Illustration of how the LLM-based hypermedia client, HyperBrain, navigates the Wikipedia environment.**

navigates across Wikipedia pages, assisting users in finding information about their topics of interest. Our system is not limited to a specific LLM; however, during evaluation, we used the gpt-3.5-turbo and GPT-4 models (with the *temperature* hyperparameter set to 0.9), as these models are capable of processing the large number of tokens commonly found in Wikipedia pages.

### 3.2 Scenario: Navigating Wikipedia

Wikipedia is a great testing environment for HyperBrain, since its entries commonly offer many hyperlinks accompanied by highly expressive context in natural language. At the same time, the diversity of topics in a Wikipedia entry poses an interesting challenge in

deciding on the most suitable hyperlink among all the hyperlinks available.

In the Wikipedia environment, each session with HyperBrain starts with the initialization of a keyword representing a topic of interest. This keyword indicates the Wikipedia entry to which the client needs to navigate. Additionally, an entry point, specified as the URI of the current Wikipedia entry, is provided. For our experiments, we selected “Lion” and “Genus” as keywords, and the locations of the Wikipedia entries for “Zurich” and “Rome” in the English Wikipedia<sup>4</sup> as entry points.

Initially, the client performs a first query (as shown in Lst. 1), prompting the LLM to *guess* whether the entry point corresponds to the hypermedia reference that identifies the Wikipedia entry related to the given keyword. If the LLM considers that the entry point does not correspond to the keyword, the client proceeds by obtaining the context (i.e., the current page). This includes executing an HTTP GET request to retrieve the current Wikipedia entry in HTML format, and then extracting the hypermedia references (e.g. <https://en.wikipedia.org/wiki/Italy>) and their associated titles (e.g. “Italy”). Here, we limit the search to the first 200 available links within paragraphs, and ignore certain hypermedia references that do not relate to the topic documented in the Wikipedia entry (such as the hypermedia reference “/wiki/Wikipedia:About”). This decision derives from the structure and content of the Wikipedia environment (but it possibly limits the client’s effectiveness in other environments).

After obtaining the application context, another query is submitted (as shown in Lst. 2), prompting the LLM to *guess* the most suitable hypertext for transitioning the application state closer (in terms of the Wikipedia hypergraph) to the Wikipedia entry associated with the keyword. The suggestions are accompanied by a textual explanation, providing insights into why a particular hypertext is being recommended. HyperBrain then follows the recommended hyperlink. These steps (querying about goal achievement and hyperlink recommendations, gathering the context, and following links) are repeated until the first query returns a result indicating that the visited Wikipedia entry likely corresponds to the given keyword. The behavior of the client is depicted in Fig. 1.

Given this algorithm and queries, HyperBrain is able to navigate the Wikipedia environment. Our results provide interesting insights: First, the path identified from “Zurich” to “Genus” is not straightforward. The client navigates through the entries for “Turicum settlement”, “La Tène culture”, “Celts language”, and eventually, reaches “Genus” through “taxonomy biology”. In total, HyperBrain required five steps, which is not optimal since a shorter route could have been to traverse to “Zurich Zoo” and then, via any animal, arrive at “Genus”. However, the provided steps and explanations are reasonable. For instance, in the case of “Turicum settlement”, HyperBrain assumes that the word “Turicum” might refer to a plant or animal genus found in the area during Roman times (see Lst. 3). It hence mimics understanding of complex relationships and can search for different meanings of a word. Second, when attempting to reach “Lion” from “Rome”, the client follows an infinite navigation loop. It, first, traverses to “Roman mythology” because, as

<sup>4</sup>That is, <https://en.wikipedia.org/wiki/Zurich> and <https://en.wikipedia.org/wiki/Rome>

```

response = {
  "answer": False, "keyword": "Turicum",
  "explanation": "It is the Roman name for the city of Zurich "
               "and could potentially refer to a genus of "
               "plants or animals found in the area during "
               "Roman times."
}
response = {
  "answer": False, "keyword": "La Tene culture",
  "explanation": "The La Tene culture is a pre-Roman Iron Age "
               "culture that existed from the 5th to the 1st "
               "century BCE and is known for its distinctive "
               "art style, which included intricate geometric "
               "designs and stylized animal motifs, as well "
               "as its advanced metalworking techniques."
}
response = {
  "answer": False, "keyword": "Continental Celtic",
  "explanation": "It refers to a group of ancient Celtic "
               "languages and cultures that shared a common "
               "ancestor."
}
response = {
  "answer": False, "keyword": "Celtic languages",
  "explanation": "It refers to a group of related languages "
               "belonging to the same language family or "
               "genus."
}
response = {
  "answer": False, "keyword": "Language family",
  "explanation": "A language family is a group of languages "
               "that are related through a common ancestor "
               "language or proto-language, much like how "
               "biological organisms are classified into "
               "genera based on their evolutionary history."
}
response = {
  "answer": False, "keyword": "Taxonomy (biology)",
  "explanation": "Refers to the branch of biology that deals "
               "with the classification of organisms into "
               "groups or taxa, such as genus and species."
}
response = {
  "answer": True, "keyword": "Genus"
}

```

**Code Listing 3: HyperBrain output when tasked to navigate from “Zurich” to “Genus” on Wikipedia.**

reported by the client, lions were often depicted in Roman mythology. Next, HyperBrain repeatedly transitions among “Hercules”, “Heracles”, “Greek mythology”, and “Hera”. Finally, HyperBrain followed the optimal path to navigate from “Zurich” to “Lion”, initially traversing to “Coat of Arms” and then to “Lion”.

### 3.3 Discussion and Limitations

Our demonstrator shows that LLMs can be valuable tools to assist goal-driven exploration of hypermedia environments. Through our experiments, we gained interesting insight on how to form queries that can effectively prompt the LLM to assist in hypermedia navigation. First, concealing the user’s exact intention may be beneficial, as when the intention is too strictly expressed, the LLM behaves as if the means to satisfy it are too limited, potentially leading to fewer suggestions. In our demonstrator, we preferred receiving a low-confidence suggestion to none at all. We tested different query forms with verbs like “return”, “provide”, “deliver”, and “get”, and found that “guess” yielded the best results when determining the goal application state and querying for hyperlink suggestions. The open-ended nature of guessing, combined with the phrasing used to ask for the “most likely related” keyword, led to the most suitable responses. Second, our experiments revealed that we can adequately restrict query responses to the keywords TRUE and FALSE (see Lst. 1), which can be used directly to evaluate

the condition for terminating navigation. Finally, requesting an explanation for the suggested keywords from HyperBrain (see Lst. 2) can improve its reliability and help the user to reach their goal application state more effectively.

Our hypermedia client has demonstrated its flexibility by providing useful suggestions for navigation steps within the Wikipedia environment. However, certain implementation choices are tailored to the characteristics of this environment, which may limit the client’s effectiveness in more general hypermedia environments. Such choices concern processing HTML documents for ignoring a subset of the available hypermedia references, and the ability of the client to only execute HTTP GET requests. Additionally, the presented version of HyperBrain lacks support to dynamically construct input during interactions with a hypermedia environment<sup>5</sup>, and does not maintain a record of interactions for effectively avoiding loops in navigation. Finally, the effectiveness of the client can be hindered in cases where query results are too random or deviate from the expected structure. For example, if the LLM is unable to provide a suggestion because it cannot determine the relevance of the given keyword to the available hypertext, the client is unable to proceed with its navigation.

Further investigation is needed to validate LLM-based hypermedia guidance in various environments, including plain hypertext and more generic hypermedia settings, and for complex goals. We argue that LLMs effectively assisting navigation based on the application context *and* users’ goals would represent an advancement beyond typical search engine pages that require users to restart navigation each time they seek suggestions. Additionally, considering the abundance of models and formats for hypermedia-based interaction descriptions, we argue that LLM-based hypermedia guidance can be effective in hybrid virtual and physical hypermedia environments, where users may benefit from guidance upon interacting in more heterogeneous and large-scale settings. For instance, an LLM-based hypermedia client capable of interpreting W3C Web of Things Thing Descriptions [6] describing hypermedia interfaces of physical devices could potentially facilitate the more seamless navigation across Web-based physical environments.

## 4 CONCLUSION

In this paper, we introduced HyperBrain, an LLM-based hypermedia client that navigates hypermedia environments to achieve user goals, where an LLM provides local guidance considering users’ global goals. We demonstrated the effectiveness of our system in a scenario for exploring plain hypertext in the Wikipedia environment. We discussed our observations on LLM-based hypermedia navigation, as well as the potential of this approach to facilitate interaction in hypermedia environments with resources that represent both, virtual and physical resources. Overall, we conclude that the usage of an LLM to predict which hypermedia controls are most likely to support goal-driven hypermedia exploration is feasible and effective, and we propose that this might provide an effective interface between humans and software agents who use the Web together.

<sup>5</sup>A version of the LLM-based hypermedia client that supports the contextual generation and usage of input entries is available online: [https://github.com/Interactions-HSG/hyperbrain/blob/main/hyperbrain/objects/hyperbrain\\_cherrybot.py](https://github.com/Interactions-HSG/hyperbrain/blob/main/hyperbrain/objects/hyperbrain_cherrybot.py).

## ACKNOWLEDGMENTS

This article emerged as one of the consequences of joint discussion by two working groups at the Dagstuhl Seminar on “Agents on the Web”<sup>6</sup> held in February 2023, and from further explorations across seven research groups in different fields of Computer Science. We would like to thank the organizers of the Dagstuhl Seminar (Olivier Boissier, Andrei Ciortea, Andreas Harth, Alessandro Ricci) as well as the members of those working groups: Samuele Burattini, Brian Logan, Matthias Kovatsch, Sebastian Schmid, Andreas Harth, Daniel Schraudner, Rem Collier, Mahda Noura, Cleber Jorge Amaral, and Jean-Paul Calbimonte; it is a highly enjoyable, and a privilege, to work with you! This research was furthermore supported by the Swiss National Science Fund (HyperAgents, Project #189474), and the EU Horizon 2020 program (IntelloT, Grant #957218).

## REFERENCES

- [1] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.
- [2] R. T. Fielding and R. N. Taylor. 2002. Principled Design of the Modern Web Architecture. *ACM Transactions on Internet Technology* 2 (2002), 115–150. Issue 2.
- [3] Larry M. Masinter and Daniel W. Connolly. 2000. The “text/html” Media Type. RFC 2854. <https://doi.org/10.17487/RFC2854>
- [4] Simon Mayer. 2014. *Interacting with the Web of Things*. Ph. D. Dissertation. ETH Zurich.
- [5] Simon Mayer and David S. Karam. 2012. A Computational Space for the Web of Things. In *Proceedings of the 3rd International Workshop on the Web of Things (WoT 2012)*. Newcastle, UK.
- [6] Michael McCool, Ege Korkan, and Sebastian Käbis. 2023. *Web of Things (WoT) Thing Description 1.1*. W3C Proposed Recommendation. W3C. <https://www.w3.org/TR/2023/PR-wot-thing-description11-20230711/>.
- [7] Darrel Miller, Jeremy Whitlock, Marsh Gardiner, Mike Ralphson, Ron Ratovsky, and Uri Sarid. 2021. *OpenAPI Specification v3.1.0*. Technical Report. OpenAPI Initiative. <https://spec.openapis.org/oas/v3.1.0>.
- [8] Mark Weiser. 1999. The computer for the 21st century. *ACM SIGMOBILE mobile computing and communications review* 3, 3 (1999), 3–11.

<sup>6</sup><https://www.dagstuhl.de/en/seminars/seminar-calendar/seminar-details/23081>