

Modeling Climate Obstruction with a Neurosymbolic Architecture

Michael DeBellis^{1,*}, George Gino², Aadarsh Balaji^{1,3} and Jacob Gino^{1,3}

¹*michaeldebellis.com, San Francisco, CA, USA*

²*Arizona State University, Tempe, AZ, USA*

³*University of California, Berkeley, CA, USA*

⁴*University of Wisconsin–Madison, USA*

Abstract

Climate change poses one of the gravest threats humanity has ever faced, yet global action remains fragmented and inconsistent. Unlike prior environmental challenges such as the ozone crisis climate change has not prompted a cohesive international response. While inadequate science communication is sometimes blamed, emerging social science research from the Climate Social Science Network (CSSN) highlights a more systematic cause: a coordinated effort by fossil fuel-aligned actors to spread disinformation, a phenomenon now referred to as Climate Obstruction. This paper presents a formal ontological framework for modeling CSSN theories, grounded in a Neurosymbolic architecture that combines the deductive reasoning of OWL (Web Ontology Language) with the inductive modeling of vectors from Large Language Models (LLMs). By uniting logical inference with empirical understanding, our system enables structured representations of climate disinformation strategies and supports empirically testable models based on textual evidence and event data. The current system can retrieve examples from several heterogeneous databases in a single natural language query. It can also correctly classify new examples of Green Washing. It has not yet been tested with social scientists and that is the most important next step.

Keywords

Climate Obstruction, Climate Change, Climate Social Science Network (CSSN), LLM, OWL, Neurosymbolic, Retrieval Augmented Generation (RAG), knowledge graph

1. Introduction

Climate change poses one of the gravest threats humanity has ever faced, yet global action remains fragmented and inconsistent. Unlike prior environmental challenges such as the ozone crisis climate change has not prompted a cohesive international response. While inadequate science communication is sometimes blamed [1], recent work in the social sciences points to a more deliberate cause: an organized campaign of disinformation by powerful fossil fuel interests, a phenomenon known as Climate Obstruction [2, 3, 4]. A group of social scientists known as the Climate Social Sciences Network (CSSN) [5] are developing models of Climate Obstruction. The goal of this project is to provide a Neurosymbolic knowledge graph built on the theories and databases developed by the CSSN. Second paragraph.

This research aims to formalize key theories from the climate obstruction literature using a Neurosymbolic approach, integrating symbolic representations expressed in OWL (Web Ontology Language) with vector-based models from a Large Language Model (LLM) via a Retrieval Augmented Generation (RAG) architecture. In the short term, the goal is to support researchers by providing a unified, natural language RAG portal for retrieving and exploring documents related to climate obstruction across heterogeneous sources. More broadly, this work explores a novel method for formalizing social science theories. OWL provides a description logic foundation for checking model consistency and drawing deductive inferences, while LLM-based embeddings offer a complementary, inductive analysis of meaning.

Proceedings of the Joint Ontology Workshops (JOWO) - Episode XI: The Sicilian Summer under the Etna, co-located with the 15th International Conference on Formal Ontology in Information Systems (FOIS 2025), September 8-9, 2025, Catania, Italy

*Corresponding author.

✉ mdebellissf@gmail.com (M. DeBellis); georgegino@gmail.com (G. Gino); aadarsh.balaji@gmail.com (A. Balaji);

jacobkgino@gmail.com (J. Gino)



© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

Together, these approaches form a hybrid framework for expressing, validating, and reasoning about complex sociopolitical phenomena. The next sections will discuss how our work compares to previous research in RAG architectures and climate obstruction.

1.1. Relation to Previous Work: RAG Architecture

The RAG architecture was first utilized to answer specific types of questions across all domains. The core idea behind RAG is to replace the broad but shallow knowledge of an LLM with a narrow but deep knowledge base document corpus for a specific domain. With the rise in popularity of LLMs, the architecture was seen as a way to leverage a domain specific corpus of documents and avoid the limitations of a traditional LLM: hallucinations and black-box reasoning [6]. These early RAG systems and the majority of RAG systems to date use relational databases to store the document corpus. However, recent work has shown that there are significant benefits to using a knowledge graph rather than a relational database to store the corpus and model the domain [7, 8, 9].

Most knowledge graph RAG systems focus on entity-centric retrieval using Named Entity Recognition over lightweight knowledge graphs (e.g., Wikidata, ConceptNet, UMLS) that are semantic networks but have no formal, logical foundation as OWL knowledge graphs do. These systems are limited to using text matching to identify common patterns for basic entities such as people, organizations, and places. They employ graph traversals or learned embeddings to guide document selection. These approaches often rely on shallow graph structures or neural graph encoders and emphasize factual question answering in specific domains. In contrast, our Climate Obstruction RAG system leverages an OWL ontology with full Description Logic semantics. Rather than simple fact retrieval, our system is designed to support causal modeling, event decomposition, and an integrated model of climate obstruction theories.

Another innovation of our approach is that we model beliefs of both groups and individuals using reified triples, an approach pioneered in the Cognitive Modules ontology [10]. Modeling beliefs is inherently difficult in OWL due to OWL's logical rigor. Any realistic model of beliefs will soon face the challenge that different belief systems are logically incompatible. At the same time modeling beliefs is essential for this type of social science research, especially the concept of a Field Frame from the work of Brulle discussed below. This project re-used much of the code from a previous RAG system developed for Dental Materials (DrMo)[8] built on the AllegroGraph platform. The most important feature from AllegroGraph that we utilized to integrate with ChatGPT are AllegroGraph Magic Properties. Magic property is the AllegroGraph term for their proprietary extensions to SPARQL. Magic properties have the same syntax as standard SPARQL predicates, however they execute functions as well as pattern matching on the knowledge graph.

1.2. Relation to Previous Social Science Research

Recent research in the social sciences has made significant advances in understanding climate obstruction using computational techniques, particularly Natural Language Processing (NLP). For example, named entity recognition and approximate string matching to uncover ties between climate misinformation and philanthropic institutions [11, 12]. In [12], researchers used social network analysis and topic modeling to identify a tripartite structure within the U.S. climate change countermovement, showing how informal social networks reflect distinct ideological and industrial coalitions. While these studies effectively show specific patterns, they are not designed to represent or reason about domain theories. They are essentially individual tools designed to do one specific type of analysis in isolation.

In contrast, our work introduces a Neurosymbolic framework that not only allows for different types of NLP analysis, it provides an integrated formal OWL-based ontology with vector-based models that explicitly model the theories. This enables the system not only to analyze textual content but to represent, test, and refine theoretical constructs such as Field Frames, types of Greenwashing, and causal influence chains. Our approach complements prior NLP-driven work while expanding the methodological frontier to include symbolic representation and deductive reasoning. Our plan is

to incorporate some of these prior NLP techniques into our system. Due to the knowledge-based foundation of our work, it should be simpler to implement and integrate them, yielding useful synergies. For example, there is already a rich library in AllegroGraph for the kind of social network analysis used in [12].

1.3. Conventions and Roadmap

Throughout this paper, SPARQL queries are in Courier New 10 font. References to ontology entities are in italics. Names of Classes and Individuals are capitalized. Names of properties are in lower case. Bold is used for emphasis. The code and ontology are available via an open source license and can be found at our GitHub site [13]. The ontology documentation can be found at: https://mdebellis.github.io/Climate_Obstruction/. Additional examples can be found on our GitHub wiki [14].

The structure of the paper is: Section 2 describes the data pipeline and run-time architecture. Section 3 describes how we modeled the concepts defined in the social science literature and how that model is used. Section 4 describes next steps and conclusions.

2. Methods: Data Pipeline and Architecture

In this section we describe how we populate the knowledge base, how we generate vectors to model the meaning of the text in the corpus, and the system architecture.

2.1. Retrieval Augmented Generation (RAG)

Retrieval-Augmented Generation (RAG) architectures enhance Large Language Models (LLMs) by grounding their outputs in a curated external corpus, rather than relying on the LLM’s internal parameters [6]. This approach directly addresses two major limitations of LLMs: black-box reasoning and hallucinations.

Standard LLMs lack explicit representations of knowledge. As a result, their reasoning processes are opaque, sources cited in responses are post hoc and do not reflect the internal mechanisms by which answers were generated. It is currently impossible to trace which specific model parameters contributed to a given output [15]. This opacity also leads to hallucinations, as the model cannot assess whether its internal representations are a good match to the prompt. RAG mitigates these issues by shifting the knowledge source to a transparent, retrievable document corpus. This allows users to trace responses back to verifiable sources and provides greater control over domain knowledge. As with all architectural decisions, this comes with a trade-off: RAG systems forgo the generality of standard LLMs in favor of precision and reliability in a narrow domain.

2.2. Neurosymbolic Modeling

Neurosymbolic modeling combines vector embeddings with symbolic knowledge graphs in a unified framework [16]. Using a knowledge graph rather than a traditional relational database to store the RAG corpus offers key advantages such as explanation generation, deductive inference, and interactive graph-based exploration [8]. Ontologies enable the reuse of rigorously defined vocabularies curated by domain experts. Our system builds on four ontologies: Dublin Core for document metadata [17], the Gist Upper Ontology for general concepts [18], the Universal Moral Grammar (UMG) and the Cognitive Modules ontology for modeling agents, causality, and moral responsibility [10, 19]. We implement this architecture using AllegroGraph from Franz Inc. as the Neurosymbolic platform. Ontologies are developed in Protégé [20] and deployed to AllegroGraph, which supports tight integration with OpenAI’s API, allowing seamless use of ChatGPT for vector generation and interaction with the data in the corpus via natural language.

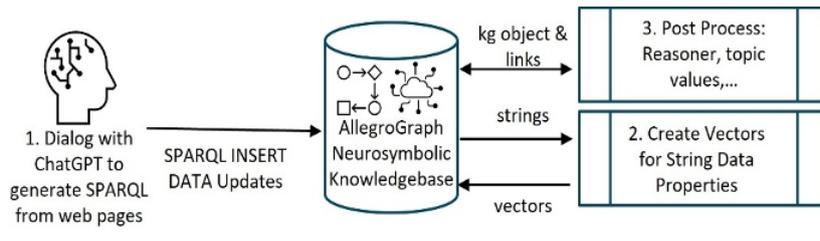


Figure 1: ChatGPT Data Pipeline

2.3. Data Pipeline

Our original data pipeline scrapes publicly available web content related to climate obstruction, including litigation records, corporate donations, and documented instances of regulatory evasion. Chrome Developer Tools are used to inspect HTML and network activity, guiding the development of custom Python scripts using the BeautifulSoup library [21]. This creates knowledge graph objects that model the corpus documents and are linked to the ontology definitions of climate obstruction theories. This pipeline is discussed in detail in [22]. This section describes the new data pipeline that supplements the previous pipeline using ChatGPT to directly transform web pages into objects in the knowledge graph.

Recently we have had great success using ChatGPT to directly transform web pages into SPARQL INSERT DATA updates. This greatly simplifies the process by removing the need to generate CSV files and transform scraped CSV files into knowledge graph objects. The memory capability of ChatGPT makes this possible. As the developer interacts with ChatGPT, it understands the structure of the ontology and it becomes easier to create a prompt and have ChatGPT generate an INSERT DATA update. This ChatGPT pipeline is shown in Figure 1. For example, the following prompt was used to add data to the knowledge graph:

For example, the following prompt was used to add data to the knowledge graph:

“I would like you to generate a SPARQL INSERT DATA update. The info comes from this page: <https://www.eenews.net/articles/every-president-since-jfk-was-warned-about-climate-change/>. I have already defined an Event called :Every_president_since_JFK_was_warned_about_climate_change. What I would like is to have sub events for that event such as JFK_Warning, LBJ_Warning, etc. using data from that page. Each new object will be an instance of the :Warning_Event class. Here is an example of the template I would like you to follow: <SPARQL INSERT DATA Example>.”

This prompt generated an INSERT DATA update that created 10 new objects. The following code is an example of the SPARQL generated by ChatGPT:

```
INSERT DATA {
  :LBJ_Warning rdf:type :Warning_Event ;
  rdfs:label "LBJ receives climate warning" ;
  gist:startDateTime "1965-01-01T00:00:00"^^xsd:dateTime ;
  gist:endDateTime "1965-01-01T00:00:00"^^xsd:dateTime ;
  :is_sub_event_of :Every_president_since_JFK_was_warned_about_climate_change ;
  rdfs:isDefinedBy "https://www.eenews.net/articles/every-pre..." ;
  skos:definition "A federal science report, \"Restoring ...\"." .}
```

Using ChatGPT significantly reduces the effort required to add data to the knowledge graph, however, it is far from perfect. To date we’ve never had the correct SPARQL update generated on the first attempt. There are always errors such as using a semi-colon where there should be a period or using the wrong name for a class. However, these are easily repaired with iteration and in addition since ChatGPT keeps long term memory for each user, it continually improves its ability to transform web pages to SPARQL.

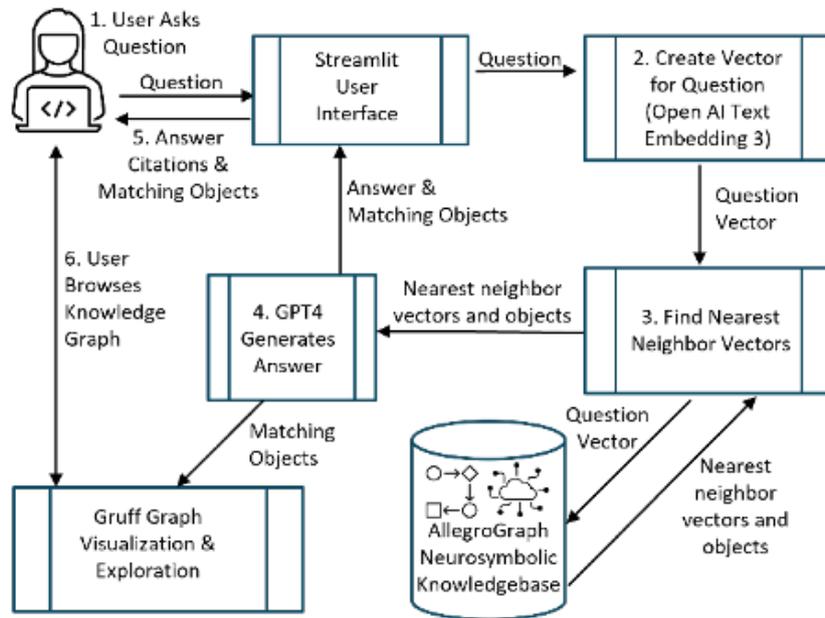


Figure 2: Runtime Architecture

After adding new objects to the graph, we perform post processing (see Figure 1). The second step is to run an AllegroGraph function invoked via the WebView user interface to generate vectors for string based properties via an API to the Open AI text-embedding-3 model. In the example above *skos:definition* is a property that has large text strings that define the meaning of entities and hence we create vectors for each new string value of that property. After generating vectors for appropriate properties, we perform post processing that generates additional knowledge-graph objects. This includes running the reasoner as well as running domain specific functions that generate other objects and object properties. For example, the AllegroGraph Free Text Index (FTI) [23] is utilized to analyze document text strings and develop has topic properties that link document objects to appropriate entities in the ontology using NLP techniques such as stemming and bag of words [24].

2.4. Run-Time Architecture

Figure 2 shows the run-time architecture. Every time the user enters a new question, the following six step process is initiated.

Step 1: User Enters Prompt. The user enters a prompt via the Streamlit User Interface (UI). Streamlit runs on the AllegroGraph Python client and generates a simple user interface (see Figure 3) with frames for the maximum number of results, the relevance threshold (a number between 0 and 1 that indicates how close two vectors must be to indicate a match), the question (prompt), the answer, and the relevant text from the supporting documents in the corpus. The Streamlit UI is powered by a SPARQL query that utilizes an AllegroGraph “magic property”. Magic property is the Franz term for their proprietary extensions to SPARQL. A magic property has a similar syntax to a SPARQL property. However, in reality it calls a function. The input parameters to the function must be bound before the property is used and are in parentheses where the object variable for a triple would be. The magic property invokes a function with those parameters and the values returned are bound to variables in parentheses where the subject of a triple would typically be.

Step 2: Create Vector for Prompt. The Streamlit UI contains the template for a SPARQL query with the magic property *askMyDocuments*. When the user enters a new prompt, that prompt along with the values for the minimum match and the number of matching documents are assembled into a SPARQL query that is passed to AllegroGraph through the Python client. The code below shows the essential structure of the query (excluding prefixes and additional optional patterns) generated by the

Climate Obstruction Portal

Enter question here:

Are there any conspiracy theories linking Covid and Climate Change?

Answer:

****Result 1:****
"The discussion revolves around the controversial idea of climate lockdowns, which some conspiracy theorists argue is a plan by elites to restrict freedoms under the guise of climate change management. These theories focus more on potential rather than actual events, diverging from established fact-checking processes. Simultaneously, there's debate on whether the climate crisis could justify offenses, pointing to the need for concrete actions like a green economic transformation to prevent extreme measures. Articles suggest that the COVID-19 pandemic has shown a glimpse of emission reduction possibilities, stressing the urgency for a green recovery focused on renewable energy investments. However, the threat of bailouts for fossil fuels looms large, indicating resistance to *nonrepressive environmental policies. Moreover, experts warn against*

[View answer graph in Gruff](#)

Supporting Documents:

Document 1:
"The climate lockdown conspiracies claim that a clandestine group of elites are planning to use climate change as a justification to enact widespread lockdowns and curtail freedoms. This conspiracy draws on a wide range of unconnected real-world events and suggests that their possibility of happening again is all the proof required. The focus on possibility instead of reality demonstrates the deep-seated philosophical differences between conspiracy theories and the fact-checking processes that counter them"

Document 2:
"Whether the climate crisis can constitute an excuse for committing an offense."

Document 3:
"The Guardian view on the climate and coronavirus: global

Figure 3: Climate Obstruction Portal UI

question in Figure 3. After askMyDocuments retrieves the top matching text chunks based on vector similarity, the query uses the reverse pattern ?doc ?prop ?content to identify the knowledge graph object from which the string came from.

```
SELECT *  
WHERE {  
  BIND("Are there any conspiracy theories linking climate change and the  
  Covid pandemic" AS ?query)  
  (?response ?score ?vec ?content) llm:askMyDocuments(?query  
  "climate_obstruction" 7 0.7) .  
  ?doc ?prop ?content .  
  OPTIONAL { ?doc :has_topic ?topic }  
}
```

Step 3: Find Nearest Neighbor Vectors. The user's prompt is passed on to the Open AI API using the askMyDocuments magic property to create a vector for the prompt. That vector is matched to existing vectors in the system using a cosine nearest neighbor function (a standard way to calculate distance in a multidimensional vector space). This is then used by askMyDocuments to find the N nearest neighbors in the vector space of the Neurosymbolic knowledge base (where N is the parameter for maximum matching strings) that are above the relevance threshold. The next triple after the magic property is used to trace back from each matching vector to the text string with that vector (bound to ?content) and the document object that has that string as a property value (bound to ?document). In addition, there are several optional triples (most not shown for brevity) to find additional relevant objects such as the author, the topic, and any objects the entity is a part of. These all must be inside OPTIONAL statements because not all entities will have a match for every triple. Hence, the SPARQL query would often fail if the additional matches weren't optional.

Step: 4 GPT4 Generates Answer. The matching vectors as well as the prompt are then passed again to the Open AI API, this time to invoke GPT4. GPT4 generates the response.

Steps: 5 and 6 Display Answer and Browse Knowledge Graph. The response generated by GPT4 is returned to the UI along with the matching text strings. These are displayed in the user interface. In

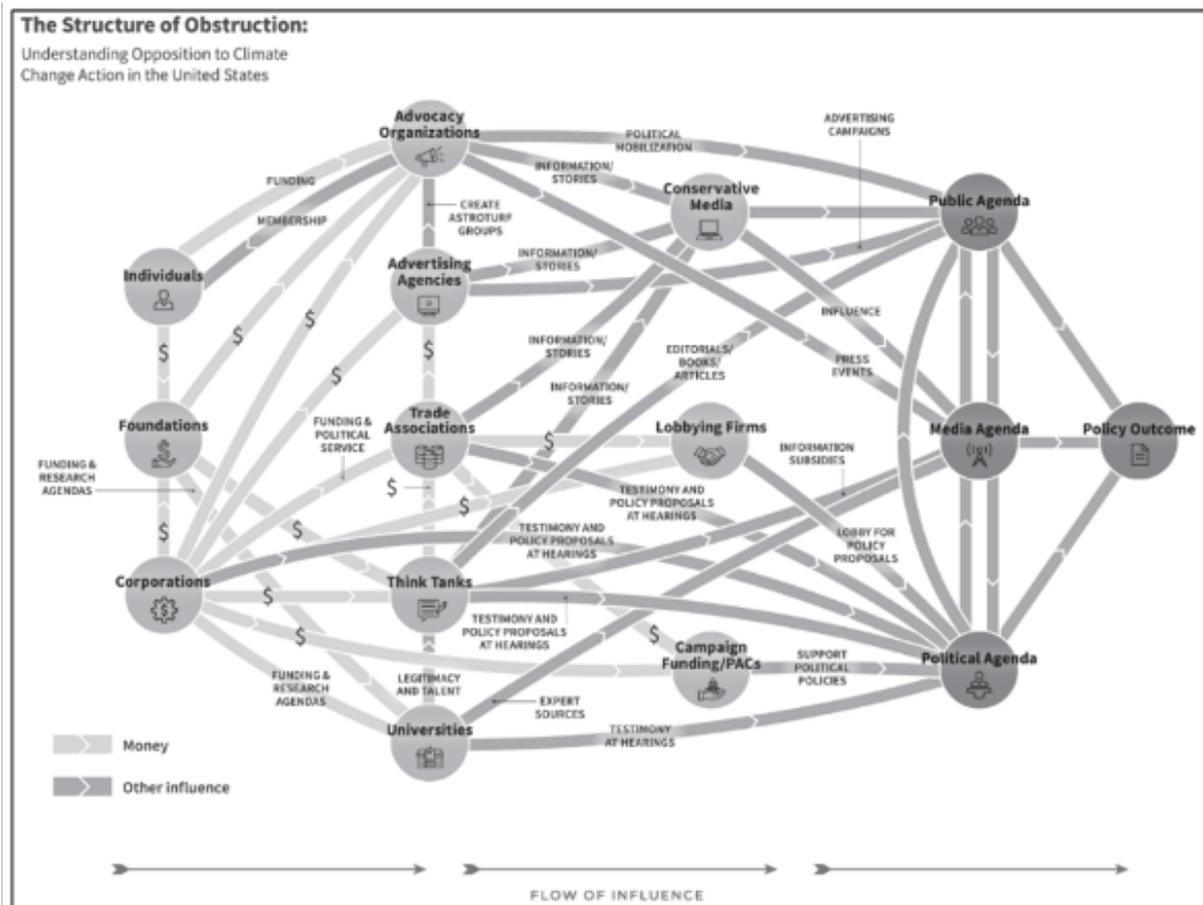


Figure 4: Influence Graph from [3] used with author's permission

addition, the SPARQL query that was generated is saved to the copy/paste buffer so that the user can select "View answer graph in Gruff" and paste the query into Gruff. Gruff is the AllegroGraph graphical browser. It takes a SPARQL query and returns all the relevant knowledge graph objects which the user can then visualize in a graph. Gruff generates a legend in the left panel where the datatype for each node (typically an OWL class) and the name of each property are color coded.

Figure 7 below show the graph for this example after the user has manipulated it to view more relevant details. The details of this graph will be explained in the next section when we discuss modeling the Climate Obstruction theories.

One of the most important additions to the system is a ChatBot implemented via the AllegroGraph chatState magic property. Unlike askMyDocuments, which processes each query in isolation, chatState supports persistent conversational memory by storing prior prompts, answers, and retrieved results in a context object that persists across turns. This enables follow-up questions, clarifications, and more abstract forms of reasoning. We currently utilize the AllegroGraph user interface for the ChatBot and will discuss one way it is used in 3.1.2.

3. Results: Building and Testing the Model

3.1. Designing the Climate Obstruction Model

Our current model of Climate Obstruction integrates three models [2, 3, 4]. These models were not designed to be part of one coherent model and one of the first examples of the value of our approach is integrating these into one logical model. In addition to providing a tool for climate obstruction researchers, a long-term goal of this project is to show how the Neurosymbolic architecture can provide



Figure 5: Influence Event Classes and Greenwashing Hierarchy

a formal foundation for theories in the social science that make falsifiable predictions. We show the first example in 3.1.2. Each of those sources emphasize different aspects of Climate Obstruction: Influence flow, Greenwashing, and Field Frames.

3.1.1. Influence Flows

The model in [2] is based on a graph of the way influence flows across stake holders in the Climate Obstruction network, this graph is reproduced in Figure 4 with permission of the author. This model is a meta-model of classes and the flows between them. We utilized the Event class to model influence flows. I.e., funding donations, testimony at hearings, support for political agendas are all examples of Events. Many of these Events are types of communication, hence subclasses of Communication Event. However, the influence model involves more than communication such as Political Mobilization. Using multiple inheritance, we were able to preserve the model where each edge is a subclass of Influence Event and most, but not all of those classes are also subclasses of Communication Event. Figure 5 (left side) shows the Influence Event hierarchy within the context of the ontology in Protégé. It also shows a class hierarchy discussed in section 3.1.2 on the right. The axioms on Create Advertising Campaign correspond to the nodes in the original diagram and illustrate how we have transformed the informal diagram into a formal model that can be analyzed and connected to real world data.

We model most of the nodes in the influence model as Agent subclasses, where an Agent can be a Person, Group, or AI Agent. The various connections in the influence model are modeled as axioms on each class as shown in the Protégé screen print on the left of Figure 5. In addition, we used the model of causality in the UMG as well as Gist properties such as giver, receiver, comes from agent, etc. to model the various properties required to completely define the influence graph in [3] that indicate the path of influence starting from corporate donations and resulting in political outcomes. Using a property hierarchy allows us to model various kinds of influence relations such as funding and publication, while at the same time recording that all these properties are different ways to causally influence other parts of the model. The top property takes the transitive closure of all causal relations. This allows us to visualize causal flows from corporate donations, to disinformation, and ultimately to political outcomes as shown in Figure 7.

3.1.2. Greenwashing

The model in [4] defines a taxonomy of types of green washing. All types of Green Washing are some type of Communication Event. Hence, the Green Washing model fit naturally under our Communication Event class as shown in the class hierarchy on the right of Figure 5. One of our recent successes has been to give the system new examples of Green Washing and ask it to classify them. For example, when given the prompt: "Delta Airlines promoted its flights as 'carbon neutral' based on the use of carbon offsets. In the lawsuit Mayanna Berrin v. Delta, the plaintiff argued these claims were deceptive, since the offsets used were of questionable environmental impact and did not guarantee real-world



Figure 6: Field Frame subclasses and instances

emission reductions. Which of the 12 greenwashing types best applies to this case? Justify your answer briefly.” The system replies: “Based on the details provided, this case can be classified as Green Washing by Impact, specifically under the sub-type of Empty Claims or Falsehoods. Here’s why: The claim that Delta Airlines’ flights are ‘carbon neutral’ relies on carbon offsets whose environmental impact is questionable and does not guarantee real-world emission reductions. This aligns with the definition of Empty Claims or Falsehoods because Delta’s assertions about carbon neutrality do not hold up under scrutiny, suggesting that the impact claimed (i.e., carbon neutrality) is unsubstantiated. The issue here is primarily that the carbon offsets in question may not provide the environmental benefits promised, rendering Delta’s claim potentially deceptive.”

3.1.3. Field Frames

Finally, we integrated the model in [3]. This model is built on the concept of a Field Frame which Brulle reused from previous work by other social scientists. Brulle defines a field frame as: “a shared perspective of the situation. [that] forms a taken-for-granted reality and defines norms for regularized patterns of social interaction” [2]. This definition was a natural fit to the concept of a Belief and a Belief System from the Cognitive Modules ontology. A Belief is holding one or more propositions to be true. We model a Proposition as a reified triple. This enables us to model the fact that different groups and individuals may believe propositions that are logically incompatible with each other without causing the ontology to be inconsistent. A Belief System is a collection of one or more Beliefs that reinforce each other and create a specific perspective by which to interpret the world. It is thus a superclass for Field Frame. Figure 6 shows the current subclasses of Field Frame and an instance of the Conspiracy Theory field frame modeling the QAnon Field Frame.

3.1.4. Using the Unified Climate Obstruction Model

Recall our example from section 2.4 where the user asked about conspiracy theories linking climate change and the Covid pandemic. The original graph returned with that query was useful. However, by a few simple actions such as expanding nodes based on specific properties and using the tree layout option in Gruff, we were able to produce the graph in Figure 7 that illustrates how the generation and results of this Field Frame follow the influence flow model in [3]. This graph shows the complex process that resulted in what is known as the “Climate Lockdown Conspiracy” [25]. During the Covid pandemic, certain authors pointed out the unintended benefits that the lockdown had on CO2 emissions in mainstream media outlets such as the Guardian. This was followed by an article that warned we must address climate change or risk resorting to draconian measures such as permanent lockdown. These rational discussions were picked up by what Brulle calls the Climate Change Counter Movement (CCCM) [3] and distorted into conspiracy theories that the goal of people advocating for climate change action was to curtail civil liberties. This chain of events is exactly the type of model described in Figure

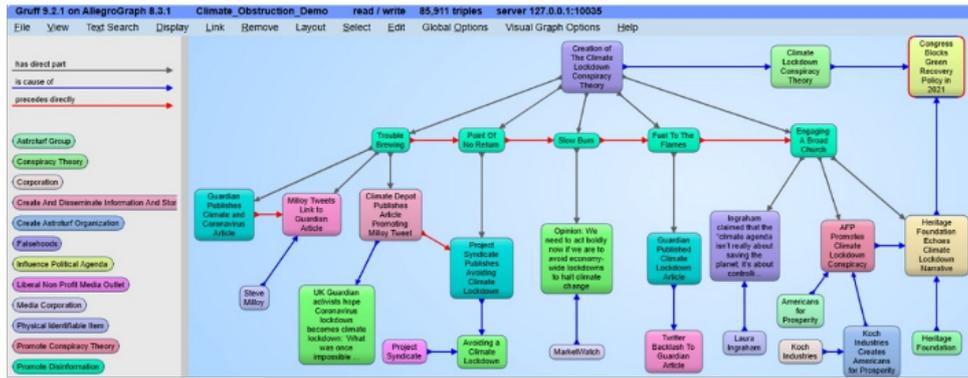


Figure 7: Object from example prompt expanded by user

5. The top level object in Figure 7 is an Event that is an instance of one of the subclasses of Green Washing. This event consists of multiple sub-events which themselves are further decomposable into the Agents, causal links, and time relations in the model. The causality connections are shown by the cause of relations in blue, the part-whole relations by has direct part properties in grey, and temporal relations by directly precedes relations in red.

In addition to modeling examples of influence flows supported by data, the URL for every document in the Corpus and for all social science concepts is stored in the knowledge graph and the user can click on any node in a Gruff graph and visit the web site, the source document, etc. using one of the Gruff menu options.

4. Discussion: Limitations and Future Work

4.1. Limitations: Tradeoffs Between RAG and LLM

While the RAG architecture offers important advances in transparency and theory formalization over LLMs, it also introduces limitations. The most obvious benefit is that the RAG architecture eliminates hallucination by grounding responses in retrieved, trusted sources. A hallucination is not simply a wrong answer, it is a phenomenon that is specific to the architecture of LLMs. Because an LLM has no explicit knowledge representation [15], it can't evaluate its capability to answer a question. In addition, due to its ability to present responses in language that is at the level of an educated human, the errors of an LLM can be very convincing even to experts [26]. In addition, an LLM has been trained on a broad range of data from academic journals to social media conspiracy theories. Thus, it may have information that is relevant but not grounded in serious research or fact based journalism. A RAG eliminates these issues because the RAG system does have access to its knowledge and can evaluate if the text in its corpus is relevant enough to the question to justify a response. This is what the similarity threshold is for. It sets the bar for comparing vectors in the knowledge base to the vector for the question and determining if there is one or more vectors in the knowledge base close enough to be used for an answer. This eliminates the hallucination phenomenon. It of course does not eliminate errors. It is still possible that a journal article or newspaper report can be wrong, but these type of "Garbage In Garbage Out" errors apply to any system. Hallucinations are unique to LLMs and eliminated by the RAG architecture. As with all architectural decisions, this benefit comes at a cost. Namely narrowing the RAG system to a specific domain. A general LLM can provide you with a good recipe for chocolate chip cookies, the Climate Obstruction portal can't. This tradeoff reflects a fundamental design choice: we prioritize transparency, testability, and alignment with formal theory over broad domain coverage. When developing a tool to support knowledge workers who require curated, high quality knowledge sources, this trade-off is usually an obvious benefit of RAG over LLM.

4.2. Limitations: Limited Integration between the LLM and the Ontology

It is somewhat ironic that a goal of OWL was to provide knowledge in a way that was accessible to humans and machines but that the format meant to be most useful to machines (formal logic modeled as an RDF graph) is actually more difficult for the LLM to utilize than natural language. At the present LLMs can't utilize knowledge in an ontology modeled as logical axioms. We currently deal with this by providing detailed text strings for classes and other entities as the value of `skos:definition` properties and generating vectors for those text strings. A more promising direction for future work is to embed the semantics of the ontology itself, not just lexical labels, into the LLM vector space through a technique known as semantic embedding. While most vector-based retrieval systems rely on language-only embeddings derived from textual definitions, semantic embedding methods like `OWL2Vec*` [27] go further: they encode the logical and structural content of an ontology into vectors. This has the effect of translating the formal semantics of OWL into a representation usable by the neural networks that define the LLM. This enables models to reflect not just surface text similarity but logical similarity. By integrating these richer semantic vectors, a RAG system can go beyond keyword matching and provide deep integration between the logical model of the ontology and the neural networks that define the LLM.

4.3. Limitation: Lack of Controlled Tests by Domain Experts

The current system is a prototype that runs on an individual machine. However, the architecture of AllegroGraph and the Streamlit user interface allow us to host the system on the Internet with trivial changes. Essentially changing a variable from `localhost` to an Internet server. That is the most important next step. As demonstrated in 3.1.2 the system can automatically classify new examples of Greenwashing. However, these were not an adequate test as the classification predictions were created by the developers rather than social scientists. As can be seen by the screen prints in this paper and the many additional examples on the project wiki [14], the system handles many types of questions and provides significant additional information via the knowledge graph. Our most important next step is to get feedback from social scientists regarding the usability of the current system, additional functionality and additional sources for the corpus. The work described in this paper was all done by volunteers with no funding. That is the only constraint to hosting the system on the Internet: to get some modest funding to support a hosting service such as Amazon Web Services.

Acknowledgments

This work was conducted using the Protégé ontology editor from Stanford University. Thanks to Franz Inc. for their support with the AllegroGraph graph database. Thanks to Dr. Bob Neches for review and feedback on the project. Thanks to Dr. Robert Brulle for providing Figure 5 and answering our many questions about the work and concepts of the CSSN.

Declaration on Generative AI

The authors used ChatGPT to assist with research and gather data only, not for writing. The authors reviewed and edited the final version of the paper and take full responsibility for its content.

References

- [1] J. Sterman, Communicating climate change risks in a skeptical world, *Climatic Change* 108 (2011).
- [2] R. J. Brulle, Advocating inaction: a historical analysis of the global climate coalition, *Environmental Politics* 32 (2022) 185–206.
- [3] R. J. Brulle, J. T. Roberts, M. C. Spencer, et al., *Climate Obstruction Across Europe*, Oxford University Press, New York, New York, 2024.

- [4] Climate Social Sciences Network, Ccssn special projects: Greenwashing tool, <https://cssn.org/special-projects/greenwashing-tool/impact/>, 2022. Accessed 19 February 2025.
- [5] Institute at Brown for Environment and Society, Climate social science network (cssn), <https://cssn.org/>, 2025. Accessed 11 June 2025.
- [6] H. Li, Y. Su, D. Cai, Y. Wang, Lemao, A survey on retrieval-augmented text generation, <https://www.semanticscholar.org/paper/A-Survey-on-Retrieval-Augmented-Text-Generation-Li-Su/e6770e3f5e74210c6863aaeed527ac4c1da419d7>, 2022. Accessed 15 April 2024.
- [7] L. Xu, L. Lu, M. Liu, Nanjing yunjin intelligent question-answering system based on knowledge graphs and retrieval augmented generation technology, *Heritage Science* 12 (2024).
- [8] M. DeBellis, N. Dutta, G. Gino, A. Balaji, Integrating ontologies and llms to implement retrieval augmented generation (rag), *Applied Ontology* (2024).
- [9] B. Peng, Y. Zhu, Y. Liu, X. Bo, H. Shi, C. Hong, Y. Zhang, S. Tang, Graph retrieval-augmented generation: A survey, arXiv preprint arXiv:2402.11922, 2024.
- [10] M. DeBellis, Modeling cognitive modules with the web ontology language: A functional architecture of the mind, in: CAOS 2023 9th Joint Ontology Workshops, Sherbrooke, Québec, Canada, 2023.
- [11] J. Farrell, The growth of climate change misinformation in us philanthropy, *Environmental Research Letters* 14 (2019).
- [12] G. Hall, L. Loy, R. J. Brulle, K. Schell-Smith, M.-M. Hu, S. Trollback, Where ideology meets private interest: the three-part composition of climate obstruction in the united states, *Environmental Research Communications* 6 (2024).
- [13] M. DeBellis, G. Gino, J. Gino, A. Balaji, Climate obstruction github repository, https://github.com/mdebellis/Climate_Obstruction, 2025. Accessed 19 May 2025.
- [14] M. DeBellis, Climate obstruction wiki, <https://tinyurl.com/climate-obstruction-wiki>, 2025. Accessed 11 June 2025.
- [15] N. Nanda, et al., Fact finding: Attempting to reverse-engineer factual recall on the neuron level, <https://www.alignmentforum.org/posts/iGuwZTHWb6DFY3sKB/fact-finding-attempting-to-reverse-engineer-factual-recall>, 2023. Accessed 26 December 2024.
- [16] A. Sheth, K. Roy, M. Gaur, Neurosymbolic ai – why, what, and how, *IEEE Intelligent Systems* (2023).
- [17] The Dublin Core Metadata Initiative (DCMI), Dcmi metadata terms, <https://www.dublincore.org/specifications/dublin-core/dcmi-terms/>, 2024. Accessed 16 April 2024.
- [18] P. Blackwood, A brief introduction to the gist semantic model, <https://www.semanticarts.com/a-brief-introduction-to-the-gist-semantic-model/>, 2020. Accessed 19 February 2025.
- [19] M. DeBellis, A universal moral grammar (umg) ontology, in: SEMANTiCS 2018 – 14th International Conference on Semantic Systems, Amsterdam, 2018.
- [20] M. Musen, The protégé project: A look back and a look forward, *AI Matters. Association for Computing Machinery SIGAI* 1 (2015) 4–12.
- [21] L. Richardson, Beautiful soup documentation, <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>, 2007. Accessed 17 April 2024.
- [22] M. DeBellis, G. Gino, A. Balaji, J. Gino, Using retrieval augmented generation (rag) and knowledge graphs to understand climate obstruction, in: IEEE Global Humanitarian Technology Conference, Boulder, 2025.
- [23] Franz Inc., Allegrograph freetext indexing, <https://franz.com/agraph/support/documentation/current/text-index.html>, 2023. Accessed 27 April 2023.
- [24] S. Bird, E. Klein, E. Loper, *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*, O'Reilly, Sebastapol, CA, USA, 2009.
- [25] E. Maharasingam-Shah, P. Vaux, *Climate lockdown and the culture wars*, Institute for Strategic Dialogue, London, UK, 2021.
- [26] R. Moore-Colyer, Ai hallucinates more frequently as it gets more advanced, <https://shorturl.at/jdZnP>, 2025. Accessed 24 July 2025.

- [27] J. Chen, P. Hu, E. Jimenez Ruiz, O. M. Holter, D. Antonyrajah, I. Horrocks, Owl2vec*: embedding of owl ontologies, *Machine Learning* 110 (2021) 1813–1845.