Using Retrieval Augmented Generation and Knowledge Graphs to Understand Climate Obstruction

Michael DeBellis
Independent Consultant
michaeldebellis.com
San Francisco, CA, USA
https://orcid.org/0000-0002-8824-9577

George Gino
Computer Science
Arizona State University
Cupertino, CA, USA
georgejgino@gmail.com

Aadarsh Balaji
Computer Science
University of California, Berkeley
New York, NY, USA
aadarsh.balaji@gmail.com

Jacob Gino
Computer Science
University of Wisconsin-Madison
Madison, WI, USA
jacobkgino@gmail.com

Abstract—**Climate change is one of the most serious crises the human race has faced. Unlike previous crises such as the destruction of the ozone layer, the world has not addressed the issue. This has been blamed on poor science communication. Social scientists have hypothesized that the primary problem is that large corporations with vested interests have orchestrated a campaign of disinformation called Climate Obstruction. This project collects various resources about Climate Obstruction into a knowledge graph corpus using a Large Language Model (LLM) as the user interface via a Retrieval Augmented Generation architecture (RAG). The corpus includes databases about Climate Obstruction as well as documents that define concepts such as Green Washing. The immediate benefit is a tool that facilitates accessing heterogeneous data in a unified portal via natural language. Our approach uses the logical foundation of the Web Ontology Language (OWL), coupled with the LLM capability to model text meaning as vectors. This enables defining rigorous climate obstruction models that can be tested against data. The current system is a prototype that is robust and usable but has not yet gone through user testing.**

Keywords — climate obstruction, Retrieval Augmented Generation (RAG), knowledge graph, Web Ontology Language (OWL), Large Language Model (LLM)

I. Introduction

Climate change is one of the most serious crises the human race has ever faced. However, unlike previous crises such as the destruction of the ozone layer, the world has not come together to address the issue. At times this has been blamed on poor science communication [1]. However, social scientists have theorized that the problem is Climate Obstruction: large corporations with vested interests in fossil fuels have orchestrated a campaign of disinformation and obfuscation [2].

The goal of this research is to develop a Neurosymbolic model [3] of the theories defined in [2], [4], and [5] (we will call these our **primary sources**). Neurosymbolic models integrate two techniques to represent meaning: logic-based symbolic representations and vector spaces created via machine learning. We use the Web Ontology Language (OWL) for the former. We use LLMs for the latter.

There are several reasons such a model benefits researchers. In the short term, it offers a central portal to find documents related to climate obstruction from diverse sources via natural language queries. That is the primary goal of this phase of the project.

The longer term goal is to demonstrate a new way to rigorously define social science models. The Description Logic of the Web Ontology Language (OWL) provides a formal model of the Climate Obstruction model. Such a model has many benefits over a model only defined with words. We can use OWL reasoners to prove that there are no logical errors in the model and to infer additional data based on the logical axioms. In addition, the ability of an LLM to model meaning of text as a vector space offers a completely different type of analysis using statistical probabilities to model the semantics of natural language.

The next section describes the concepts, methods and tools we utilized. That section progresses from first defining the general models and architecture to a concrete description of our implementation. Section III describes our current results. Section IV describes lessons learned and plans for the future.

Throughout this paper, SPARQL queries are written in `Courier New 10 font`. References to semantic concepts 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 [6]. In addition, space limitations required we limit the figures. Additional examples of prompts, answers, views of the ontology, links to source, etc., can be found on our GitHub wiki: https://tinyurl.com/climate-obstruction-wiki.

II. Methodology

A. Retrieval Augmented Generation (RAG)

Retrieval Augmented Generation (RAG) architectures allow systems to utilize the semantic embedding and NLP understanding and generation of an LLM, with a curated

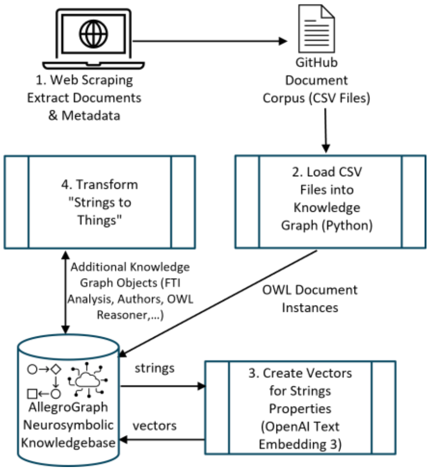


Fig. 1. RAG Data Pipeline

corpus of documents as the knowledgebase rather than the LLM's default neural network [7]. This addresses the two most significant issues with LLMs: black-box reasoning and hallucinations. A standard LLM does not know what it knows. Understanding an LLM's reasoning isn't merely difficult, as of 2025 it is an unsolved problem [8]. The lack of explicit knowledge representation in LLMs is the cause for both problems. Black-box reasoning results because, even if an LLM can find sources to support its conclusions, those sources are post-hoc rationalizations. They do not reflect how the system actually arrived at a conclusion. As demonstrated by [8], the specific cells and values that resulted in an LLM response are simply not accessible. This is also the cause for hallucinations. A standard LLM has no way to evaluate its response's quality because it has no access to the parts of the neural network used to generate a response. Hence, it has no way to evaluate whether that data was a good match for the question. The RAG architecture solves both these problems by substituting a curated corpus of documents for the domain knowledge of the LLM. Of course, as with any architectural decision, there is a trade-off. RAGs give up range in return for quality and accuracy. LLM's have incredible breadth of knowledge. The curation that ensures quality necessarily limits RAG systems to focus on one domain.

B. *Neurosymbolic Modeling*

Neurosymbolic modeling refers to the integration of embedded vectors with knowledge graphs in one unified database [3]. Utilizing a knowledge graph rather than a relational database for the corpus of a RAG system provides extra capabilities such as explanations, deductive reasoning, and graphical browsing [9].

Utilizing ontologies enables reuse of existing vocabularies offering predefined models for specific problems that have gone through a rigorous process of definition and review from leading experts on the specific domains. Our ontology is built primarily on three vocabularies. The Dublin Core vocabulary [10] provides document meta-data. The Gist Upper Model provides basic concepts such as events, places, and organizations [11]. The Universal Moral Grammar (UMG) [12] defines agents, causality, and moral responsibility.

Our system utilizes the AllegroGraph database from Franz Inc. as our Neurosymbolic platform. We use the Protégé ontology editor to define our ontology. We utilize AllegroGraph's integration with the Open AI API and the AllegroGraph Python client to integrate ChatGPT as our LLM.

C. *Data Pipeline*

A Retrieval-Augmented Generation (RAG) system takes advantage of the modular design of modern LLMs. Although LLMs appear monolithic from the outside, in practice they are composed of multiple services that can be accessed and combined independently. This architecture allows developers to use components such as vector embedding, semantic similarity search, and natural language generation as independent microservices. It is this decomposition that makes RAG possible: instead of relying solely on the LLM's internal training data, the system augments it by retrieving relevant external content based on vector similarity. For example, one common service transforms text into dense vector representations in a high-dimensional space, enabling semantic comparison with other texts. The data pipeline is displayed in Fig. 1. The data pipeline creates the vector database that is the core resource that enables the RAG architecture and hence must be run before the RAG can be used.

The first step in the data pipeline is scraping data from public web sites and databases on Climate Obstruction. All information was collected from publicly accessible pages. We used the browser-based Google Chrome Developer Tools to analyze the structure of target web pages and identify relevant content for automated collection. Specifically, we used the "Elements" tab to inspect the HTML structure of pages and the "Network" tab to observe how data was loaded dynamically. Based on this inspection, we developed Python scripts using the requests library -- a popular tool for sending HTTP requests and retrieving web content -- and the BeautifulSoup Python library to parse and extract structured data such as tables, headings, and links [13].

The current system has pages from several online databases related to Climate Obstruction. These databases include histories of litigation related to false advertising, litigation related to ignoring regulations or damaging the environment, histories of donations from corporations to think tanks and other organizations as well as the services provided in return. These databased can be utilized by social scientists to demonstrate examples of climate obstruction supported by hard data and objective litigation. These include:

- The ASA and CAP code databases: https://www.asa.org.uk/. ASA and CAP are industry regulators who can impose serious fines and other penalties on companies guilty of false advertising. We have scraped all of their data related to climate obstruction. For all scraped data we include the source page in the metadata recorded in the knowledge graph so that users can easily trace back to the original source.

- The Climate Change Litigation Database: https://climatecasechart.com/. This database is maintained by the Columbia Law School and has lawsuits against corporations, individuals, and governments from the entire world that relate to

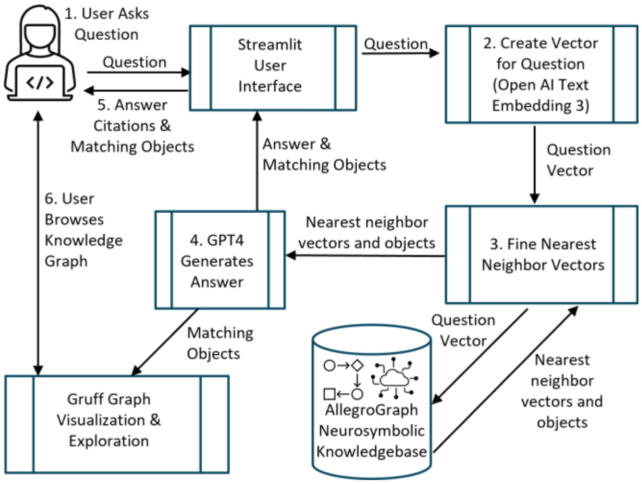

1. User Asks Question
Question
5. Answer Citations & Matching Objects
Streamlit User Interface
Question
2. Create Vector for Question (Open AI Text Embedding 3)
Question Vector
Answer & Matching Objects
6. User Browses Knowledge Graph
4. GPT4 Generates Answer
Nearest neighbor vectors and objects
3. Fine Nearest Neighbor Vectors
Matching Objects
Question Vector
AllegroGraph Neurosymbolic Knowledgebase
Nearest neighbor vectors and objects
Gruff Graph Visualization & Exploration

Fig. 2. RAG Runtime Architecture

crimes related to climate change and the environment.

- The DeSmog databases: https://www.desmog.com/databases/ This site has databases that track donations, disinformation, lobbying, and other activities related to climate obstruction.
- In addition, we have collected approximately 50 papers, web pages. book summaries, newspaper articles, and other sources.

Data from these sources were scraped and stored in CSV files. Each row in a CSV file corresponds to a document. The columns are primarily document headers that map to data properties in the ontology such as *abstract*, *introduction*, and *conclusion*. In step two, we transform the CSV files into knowledge graph objects via a direct transformation from column headings in the CSV files to data properties in the knowledge graph. The result is many new OWL instances that model documents with string data properties.

In the third step, we create vectors for the string data properties of the corpus using the Open AI API to access its Text Embedding 3 model. These vectors model the meaning of each string in a form the LLM can utilize when creating answers to questions.

In step four, we post-process objects created in step two, turning certain strings into objects and links to objects that we create or find. This process is known as “strings to things” [14]. For example, we use the AllegroGraph Free Text Index (FTI) [15] to analyze text strings from the corpus and extract references to conceptual entities such as organizations, places, events, etc. The FTI uses basic NLP tools such as stemming and bag of words [15] to find matches from small phrases and map them to the labels of entities in the knowledge graph. Such references create new *has topic* property values. Another example is transforming the author strings from the meta-data for articles to objects. For this transformation we utilize the Nameparser python library [16]. Nameparser allows us to parse through all the author names and separate them by first and last name. After the names are parsed and sorted, we search to see if an author with the same first and last name exists in the knowledge graph. If one does then we add a property called *has author* that points from the document to the author object. If one doesn’t exist it is created. We run the reasoner after transforming all the strings which adds additional data. E.g., in the case of authors it adds the inverse value *is author of* that points from the author to the document.

D. Run-time Architecture

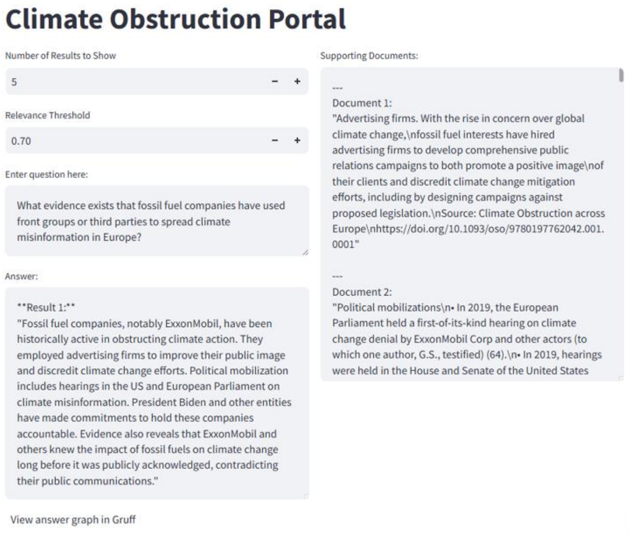
Fig. 2. shows the run-time architecture. The user enters questions via the Streamlit User Interface (UI). The user enters a question that is passed on to the Open AI API to create a vector for the question. That vector is matched to existing vectors in the system using a cosine nearest neighbor function [17]. This is passed to the Open AI API using the AllegroGraph “magic property” llm:askMyDocuments. Magic property is the term that Franz uses for proprietary extensions to SPARQL that look like SPARQL properties but in reality call functions. The llm:askMyDocuments magic property finds the vectors in the Neurosymbolic knowledge base that are within the parameters specified as part of the SPARQL query. One parameter determines the maximum number of answers. This must be an integer. The default is five. The other is a floating point number that determines how close a match counts as a good match. This must be a floating point number between zero and one. The default is 0.7. Both of these parameters can be specified in the Streamlit UI.

When the user submits a new question, it is passed through a series of functions which take a SPARQL template and fill in the template with the appropriate parameters from the UI. For example, if the user enters the question: “What evidence exists that fossil fuel companies have used front groups or third parties to spread climate misinformation in Europe?” The code excerpt below shows the essential code from the SPARQL query generated from this input with the user-defined parameters for number of matching documents and required relevance.

```
SELECT  *

WHERE {bind ("What evidence exists that fossil
fuel companies  have used front groups or
third parties to spread climate misinformation
in Europe?" as ?query)

(?response ?score ?vec ?content)
llm:askMyDocuments

(?query "climate_obstruction" 5 0.7).

OPTIONAL{?doc :has_Topic ?topic}

OPTIONAL {?author  :is_author_of ?doc}}
```

This query is generated based on the user input (the question and the parameters for relevance threshold and number of answers). The generated query is then executed using the AllegroGraph Python client. The llm:askMyDocuments magic property does the following:

1. Utilizes the Open AI API to generate a vector for the user’s question.



Climate Obstruction Portal

Number of Results to Show

5 − +

Relevance Threshold

0.70 − +

Enter question here:

What evidence exists that fossil fuel companies have used front groups or third parties to spread climate misinformation in Europe?

Answer:

Result 1:
"Fossil fuel companies, notably ExxonMobil, have been historically active in obstructing climate action. They employed advertising firms to improve their public image and discredit climate change efforts. Political mobilization includes hearings in the US and European Parliament on climate misinformation. President Biden and other entities have made commitments to hold these companies accountable. Evidence also reveals that ExxonMobil and others knew the impact of fossil fuels on climate change long before it was publicly acknowledged, contradicting their public communications."

View answer graph in Gruff

Supporting Documents:

Document 1:
"Advertising firms. With the rise in concern over global climate change,\nfossil fuel interests have hired advertising firms to develop comprehensive public relations campaigns to both promote a positive image\nof their clients and discredit climate change mitigation efforts, including by designing campaigns against proposed legislation.\nSource: Climate Obstruction across Europe\nhttps://doi.org/10.1093/oso/9780197762042.001.0001"

Document 2:
"Political mobilizations\n• In 2019, the European Parliament held a first-of-its-kind hearing on climate change denial by ExxonMobil Corp and other actors (to which one author, G.S., testified) (64).\n• In 2019, hearings were held in the House and Senate of the United States

Fig. 3. User Interface

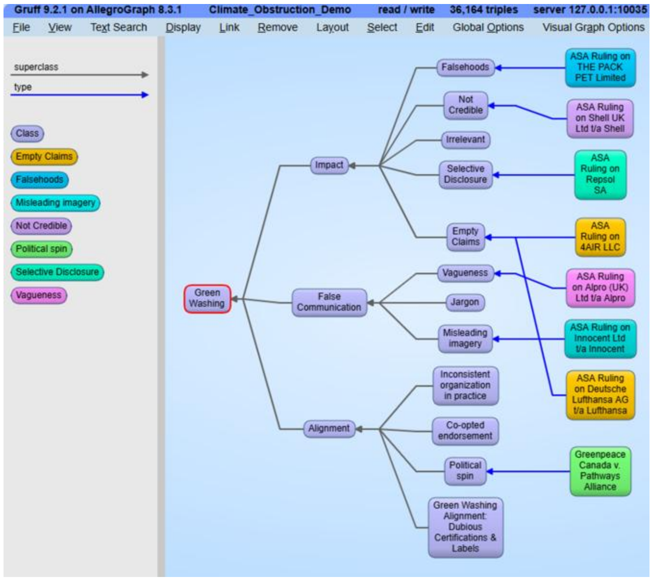


Fig. 4. Knowledge Graph Objects for Green Washing Prompt

2. Uses the cosine nearest neighbor function to find the nearest neighbors in the vector store for the repository that have a match value of (in this example) of 0.7 or higher.

3. If multiple text strings are over the match threshold takes the N number of strings that have the highest match number, where N is the parameter for maximum number of documents. In this example N = 5.

4. Sends the question, vectors, and N matching strings to the Open AI GPT-4 model that generates the response using the strings and vectors rather than its internal knowledgebase. This is the **Augmented** in RAG. We **augment** the prompt with data from our domain specific corpus rather than utilizing the much broader but shallower knowledge of the LLM.

5. Returns each answer bound to `?response` with the matching strings bound to `?content`.

The rest of the query (some parts not shown for brevity) has various optional statements to match properties such as the authors of a document, objects that a document is a sub-part of, and any entity identified by the FTI analysis as a topic of the document. These must be optional because otherwise the SPARQL query would fail if any of the properties failed to match and all properties are not always present for every document. The query is also passed to AllegroGraph's Gruff graph visualization tool so that the knowledge graph objects associated with the specific question can be viewed graphically and can be further browsed by the user for additional information.

III. Results and Discussion

The current prototype demonstrates:

1. How the system can be a useful research tool.

2. How defining a Neurosymbolic model can help formalize theoretical concepts.

A. A Retrieval Augmented Generation (RAG) Research Tool

Fig. 3. shows an example of the user interface. The user can change the relevance and result number parameters using the first two widgets on the left. The user types a question in the third box and this triggers the SPARQL query described in Section II. The answer is displayed below the question. The text strings used to generate the answer are shown in the long frame on the right. The user can also click on the "View answer graph in Gruff" link and view the relevant objects returned along with the query results.

Fig. 4 shows the Gruff graph for a new query: "What are the different kinds of Green Washing". Fig. 4 shows the graph after the user has manipulated it. Gruff displays a color coded graph with a legend on the left that maps each node and link color to the datatype and property respectively. The user can then further expand and change the layout of the graph. In this example, the user has expanded the super-classes of *Falsehoods* to see that it is a subclass of *Green Washing* which is a subclass of *Communication Event* and ultimately a subclass of the Gist *Event* class.

In this example, the user displayed the subclass relations between the *Green Washing* class and its subclasses as well as some representative instances and then chose the "layout as tree" option. This is an example of how the knowledge graph can provide additional perspective to the LLM. The question was if there are various kinds of *Green Washing.* The system enabled the user to do more than see the answer in text but to visualize the various categories of *Green Washing* as a tree in the knowledge graph with instances of each class. The URL that is the source of each instance is always available as well via a menu option on the object in Gruff. E.g., the user can select a representative instance of various types of Green Washing and go to the original page for that instance. Our GitHub Wiki shows an example where the user selects the instance of *Selective*

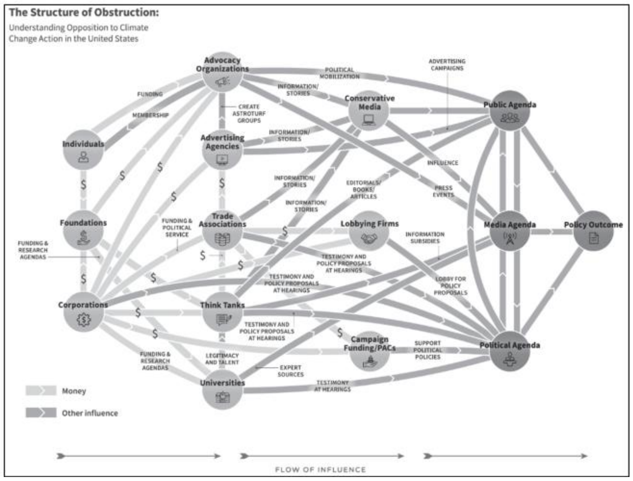


Fig. 5. Obstruction Network from [2] used with Author's permission

Disclosure: ASA Ruling On Repsol SA and selects to view the URL source for that instance.

B. Creating a Formal Model

Fig. 5 is the first figure in the introduction to [2], one of our primary sources. That chapter describes the basic model of how influence flows from various agents involved in Climate Obstruction that is the foundation for all the papers in the rest of the book. Our first task was to create a formal version of this model. We will call this model as well as the models in our other primary sources the CSSN Model.

We utilized the agent design pattern described in [12] where *Agent* is a superclass of both *Individual* and *Group*, and *Organization* is a subclass of *Group*. A *Group* differs from an *Organization* in that it has no formal control or hierarchy. E.g., "the population of America" is a *Group*, whereas "the employees of the Cato Institute" comprise an Organization. The core idea behind the *Agent* pattern is that a great deal of data and processes can refer to individuals as well as groups and organizations. They can negotiate contracts; be held morally responsible; charge for services; have emails, have addresses, etc.

We describe industries as *Groups*. Our industry model utilizes the ICB (Industry Classification Benchmark) which is the European standard for defining industry taxonomies [18]. For other types of organizations such as *Think Tanks, Foundations, Trade Associations*, etc. we followed the categories defined in [2].

Social science theories often contain terms like "beliefs," "ideologies," and "narratives," but these terms are rarely formalized in a way that supports machine-readable inference or causal analysis. To support more rigorous reasoning over such theories, we constructed a formal model that distinguishes between different levels of belief structures.

The foundation of this model is the class *Belief*, which represents any propositional attitude an agent may hold to be true. This could include simple factual beliefs (e.g., "carbon dioxide is a greenhouse gas") as well as broader ideational commitments. We then introduce the class *Belief System*, defined as a structured set of interconnected beliefs. These systems often serve as the ideological scaffolding through which agents interpret information, form preferences, and justify actions. For example, *Libertarianism* and *Environmentalism* are *Belief System*s.

Brulle [4], defines field frames as structured belief systems that help shape discourse and policy agendas in a field of contention. To model this we created a subclass of *Belief System* called *Field Frame*. *Field Frames* are shared, socially reinforced narratives that shape the strategies of *Groups*, *Organizations*, and *Individuals*.

Examples of *Field Frame* include the *Neoliberal Frame*, which emphasizes deregulation and market-based solutions and the *Climate Justice Frame*, which emphasizes equity and historical responsibility in climate discourse. Modeling these as formal classes allows the knowledge graph to capture not just the presence of these ideologies, but also their causal impact -- how they influence events, policy outcomes, and communication strategies.

This modeling approach lets us trace, for example, how a corporation's support for a think tank aligned with the *Neoliberal Frame* may lead to the publication of a report, which is then echoed in media coverage, ultimately influencing legislative inaction. By embedding these belief structures into a formal knowledge graph, we make it possible to analyze and visualize how ideologies are operationalized in real-world influence chains.

Next, we modeled the flow of *Influence* in Fig. 5. This is an excerpt of a figure from [2]. At a first glance Fig. 5 seems fairly straight forward. Most of the nodes in the graph are some kind of *Agent* and artifacts such as money and publications flow between them. However, on a closer inspection the model is more complex. E.g., concepts such as *Political Agenda* are *Field Frames*, not *Agents.* In addition, the edges in the graph are somewhat vague. After some analysis we modeled the edges as subclasses of the *Event* class in Gist. I.e., all the edges in Fig. 5 are some type of *Event.* We created a subclass of *Event* called *Communication Event.* Most of the edges in Fig. 5 are *Communication Events*. However, some such as *Provide Funding* are not. This is an example, where multiple inheritance is useful. It allows us to retain the structure of all the events that model edges in the CSSN model as subclasses of *Influence Event* while also making the classes that are *Communication Events* subclasses of that class as well. In addition we integrated the Climate Social Science Network (CSSN) Green Washing model [5] into the ontology as subclasses of *Communication Event*. Fig. 6 shows the relevant subclasses of *Event* in Protégé. We then created a sub-class of the Gist class *Artifact* called *Influence Artifact* to model things created and exchanged such as *Editorials, Books and Articles*, *Tweets*, and *Funding*. However, in some cases the event itself (e.g., *Political Mobilization*) was the example of *Influence* and there are no major artifact deliverables.

We utilized the causal model from the UMG ontology along with the ability of OWL to model super and sub properties, inverses, and transitive properties to capture the full semantics of the various definitions of the events that define the edges in the model from Fig. 5. E.g., the properties *create*, *fund*, *influence*, *publish*, and *comes from agent*, are all sub-properties

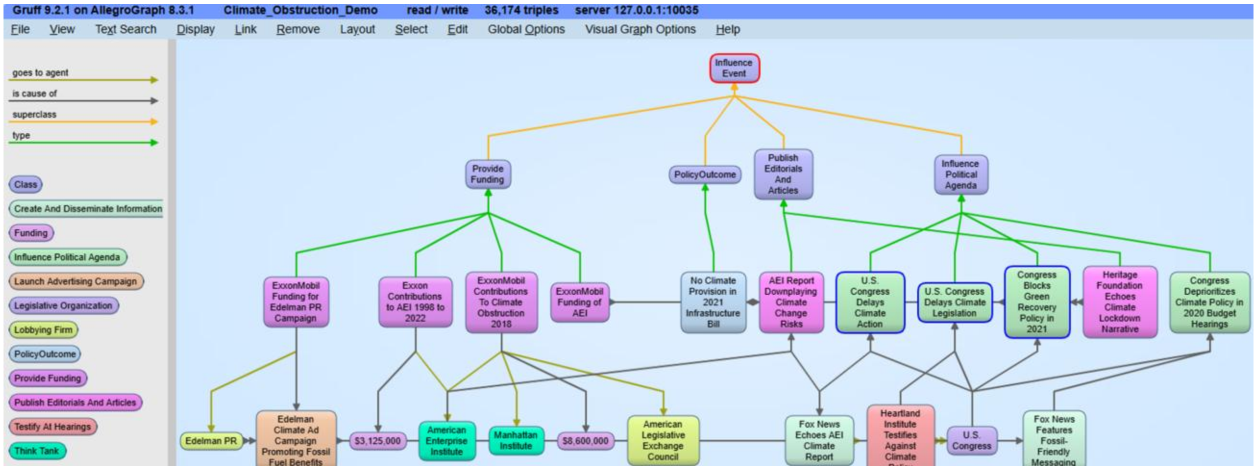

Gruff 9.2.1 on AllegroGraph 8.3.1
Climate_Obstruction_Demo
read / write
36,174 triples
server 127.0.0.1:10035
File
View
Text Search
Display
Link
Remove
Layout
Select
Edit
Global Options
Visual Graph Options
Help
goes to agent
is cause of
superclass
type
Class
Create And Disseminate Information
Funding
Influence Political Agenda
Launch Advertising Campaign
Legislative Organization
Lobbying Firm
PolicyOutcome
Provide Funding
Publish Editorials And Articles
Testify At Hearings
Think Tank
Influence Event
Provide Funding
PolicyOutcome
Publish Editorials And Articles
Influence Political Agenda
ExxonMobil Funding for Edelman PR Campaign
Exxon Contributions to AEI 1998 to 2022
ExxonMobil Contributions To Climate Obstruction 2018
ExxonMobil Funding of AEI
No Climate Provision in 2021 Infrastructure Bill
AEI Report Downplaying Climate Change Risks
U.S. Congress Delays Climate Action
U.S. Congress Delays Climate Legislation
Congress Blocks Green Recovery Policy in 2021
Heritage Foundation Echoes Climate Lockdown Narrative
Congress Deprioritizes Climate Policy in 2020 Budget Hearings
Edelman PR
Edelman Climate Ad Campaign Promoting Fossil Fuel Benefits
$3,125,000
American Enterprise Institute
Manhattan Institute
$8,600,000
American Legislative Exchange Council
Fox News Echoes AEI Climate Report
Heartland Institute Testifies Against Climate
U.S. Congress
Fox News Features Fossil-Friendly Messaging

Fig. 6. Tracing Paths of Influence with Data

of the super-property *is cause of*. The semantics for OWL properties are identical to relations in First Order Logic (FOL). In FOL relations themselves are sets of sets and one relation can subsume another relation just as a set of elements can subsume another set of elements. By making properties like *create* and *fund* sub-properties of *is cause of*, we are able to model events such as creating an astro-turf organization and the reasoner infers that the agent that created the organization is a cause of that organization. We also follow a common design pattern by having super properties to *has cause* and *is cause of* called *has cause transitive* and *is cause of transitive*. These properties are inverses as are their sub-properties and they are transitive. Thus, if we want to find all the causes of an event or artifact we can use the transitive properties. If we want to follow the causal path we can use SPARQL to find the direct (non-transitive) causal links between objects.

This new model, allows us to model causal chains of events that follow the Climate Obstruction model. Fig. 6 shows such a causal chain viewed as a hierarchical graph. At the top are several event classes corresponding to the influence types from Fig. 5: *Provide Funding, Publish Articles and Editorials, Influence Political Events*, and *Policy Outcome*. and Below them are instances of each type of event. This is an example of how formal modeling improves support for theories, in this case by using real data and by revealing the various causal and other influence connections between *Communication Events* and eventual *Policy Outcomes.* The three nodes highlighted in blue show the political influence that result from funding and other events initially launched by corporations such as Exxon. The instance of *Policy Outcome* in the center of the graph shows the ultimate result: no climate change regulations in the 2021 Infrastructure bill.

C. *Eventual Goal: Social Science with Falsifiable Predictions*

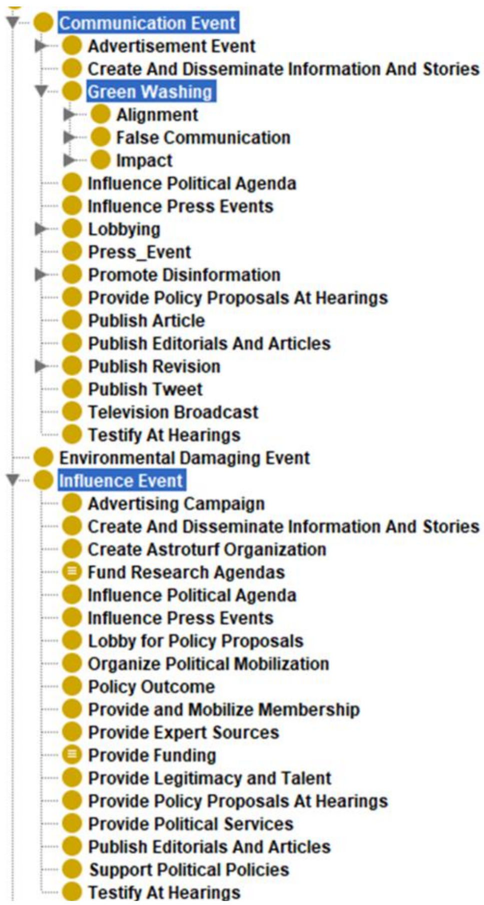
A long term goal of our research is to develop testable formal models of the CSSN Model. What we are attempting is to combine a logical model supported by deductive reasoning (the OWL reasoner) with data supported by inductive reasoning (the LLM vector space).

The social sciences are so immature compared to disciplines such as physics, that this type of model can provide value [12]. The first step for such theories is a rigorous logical model that explicitly defines the terminology, relations among the theoretical entities, and mapping to specific data points that exemplify the model. That is what we have done and in our brief amount of knowledge engineering we have already reconciled three models of Climate Obstruction; one based on Green Washing [5], the other on the Influence model described in [2] and the third based on the concept of Field Frames [4]. As part of this process, we have transformed informal models from loosely defined concepts with various interpretations to rigorous models that have been mapped to data. We have rigorously defined the models and resolved issues such as that the terms "money" and "funding" are used even though the two have the exact same meaning in the theory [19].

Most importantly, we now have access to a new type of tool in Neurosymbolic computing that provides unprecedented potential rigor coupled with a capability for falsifiable predictions. OWL is a limited implementation of logic since it implements Description Logic which is a decidable subset of First Order Logic. Many in the formal methods community would scoff at using such a limited logical language as the foundation for a formal scientific model. However, here, the power lies in integrating OWL with a large corpus of documents and to define the meaning of those documents with logic as well as a multidimensional vector space supported by hundreds of terabytes of analyzed text. This provides two major new capabilities:

1. The ability to understand natural language in a way that formal and semantic based approaches have never been able to. This provides a natural language user interface for researchers.
2. The ability to make and test predictions about the model. For example: analyze a piece of text that social scientists have categorized as a specific type of *Green Washing* and see if the model makes the same prediction.

We have successfully conducted a limited experiment using the methodology described in the latter point. Specifically, we



- Communication Event
  - Advertisement Event
  - Create And Disseminate Information And Stories
  - Green Washing
    - Alignment
    - False Communication
    - Impact
  - Influence Political Agenda
  - Influence Press Events
  - Lobbying
  - Press_Event
  - Promote Disinformation
  - Provide Policy Proposals At Hearings
  - Publish Article
  - Publish Editorials And Articles
  - Publish Revision
  - Publish Tweet
  - Television Broadcast
  - Testify At Hearings
- Environmental Damaging Event
- Influence Event
  - Advertising Campaign
  - Create And Disseminate Information And Stories
  - Create Astroturf Organization
  - Fund Research Agendas
  - Influence Political Agenda
  - Influence Press Events
  - Lobby for Policy Proposals
  - Organize Political Mobilization
  - Policy Outcome
  - Provide and Mobilize Membership
  - Provide Expert Sources
  - Provide Funding
  - Provide Legitimacy and Talent
  - Provide Policy Proposals At Hearings
  - Provide Political Services
  - Publish Editorials And Articles
  - Support Political Policies
  - Testify At Hearings

Fig. 7. Modeling Influence as Events

took an example of the *Selective Disclosure* class of *Green Washing* (the web page [20]) and took the text that defined it and utilized the AllegroGraph llm:nearestNeighbor SPARQL magic property. Note this was text that was not created by social scientists but existed on an independent web site. The nearest neighbor algorithm found the only existing instances of the class *Selective Disclosure* to be the nearest neighbors. Of course this is only one data point, not actual evidence. There is still a great deal of work to do and we offer this data point only as an example of the potential of this type of modeling.

IV. Conclusion and Recommendations

The Climate Obstruction Portal is currently a usable tool with a large corpus of data collected from a diverse set of heterogeneous sources. It is searchable via natural language with ChatGPT.

The most crucial next step is to take it from a prototype to a hosted tool on the Internet. Such a port would be trivial because the system is already completely designed as an Internet system, it is just that the server is localhost and the client and server run on the same machine. This limitation is only due to lack of funding, not to any limitations in the system. We hope to get the chance to host the tool on the Internet and collaborate with social scientists who will give us needed feedback on usability and features. One potential source is the UC Berkeley Data Science Discovery program which may provide Microsoft Azure support for project. In addition, the following are the most important next steps for future work:

Extending the corpus. We have uploaded data from many of the most important sites that are cited in Climate Obstruction papers but there is much more data to load. A related problem is transforming the string to objects is currently ad hoc. E.g., author objects and links, links from documents to topics, are each done differently. This seems like a good problem to apply Named Entity Recognition for a general solution using NLP libraries such as Spacy.

Modeling beliefs. The ontology is designed to explicitly model beliefs using reified triples. Modeling beliefs is a hard problem in OWL since different people have conflicting beliefs but a logic-based system can't handle contradictions. Reifying triples is a way to deal with contradictory beliefs. Another option is to utilize sub-graphs and put beliefs for different *Agents* into different independent sub-graphs and run the reasoner on each sub-graph. We haven't used this feature yet but plan to in the future.

Empirical validation. We have several ideas for empirical tests of the model. One was mentioned in this paper that we used for an experiment with success but we plan to do serious testing using this and other techniques that take advantage of the Neurosymbolic architecture.

The ability of the team to develop a system so rapidly illustrates the power of Large Language Models and Knowledge Graphs. The work to date has been done by volunteers working part time from February through the end of May 2025. Mr. DeBellis was primarily working on a client project and the rest of the team are undergraduates who were taking full course loads in addition to generously donating their time.

The portal is already a powerful tool for social scientists and others to organize, search and share information. The deeper potential is to formalize social science models and thereby improve their testability and validity. Both capabilities are enabled by utilizing the synergy between LLMs and Semantic Knowledge Graphs. The combination of logical modeling and deductive reasoning with modeling and reasoning based on mathematical inductive models trained on large data sets is unprecedented and has the potential to revolutionize the social sciences on a foundation of logic and data that has never before been possible.

V. Acknowledgment

This work was conducted using the Protégé ontology editor from Stanford University. Thanks to Franz Inc. for their support with the AllegroGraph database. Thanks to Bob Neches for review, guidance, and feedback. Thanks to Robert Brulle for helping us understand the Climate Obstruction model and permission to use Fig. 1 from [2].

VI. References

[1] J. Sterman, "Communicating climate change risks in a skeptical world," *Climatic Change,* vol. 108, no. 811, 2011.

[2] R. J. Brulle, J. T. Roberts, M. C. Spencer and et.al., Climate Obstruction Across Europe, R. J. Brulle, J. T. Roberts and M. C. Spencer, Eds., New York, New York: Oxford University Press, 2024.

[3] A. Sheth, K. Roy and M. Gaur, "Neurosymbolic AI -- Why, What, and How," *IEEE Intelligent Systems,* 2023.

[4] R. J. Brulle, "Advocating inaction: a historical analysis of the Global Climate Coalition," *Environmental Politics,* vol. 32, no. 2, pp. 185-206, 11 Apr 2022.

[5] Climate Social Sciences Network, "CSSN Special Projects: Greenwashing Tool," Institute at Brown for Environment and Society, 2022. [Online]. Available: https://cssn.org/special-projects/greenwashing-tool/impact/. [Accessed 19 2 2025].

[6] M. DeBellis, G. Gino, J. Gino and A. Balaji, "Climate Obstruction GitHub Repository," michaeldebellis.com, May 2025. [Online]. Available: https://github.com/mdebellis/Climate_Obstruction. [Accessed 19 May 2025].

[7] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, Q. Guo, M. Wang and H. Wang, "Retrieval-Augmented Generation for Large Language Models: A Survey," Corness University Preprints, 5 January 2024. [Online]. Available: https://arxiv.org/abs/2312.10997. [Accessed 19 February 2024].

[8] Neel Nanda et. al., "Fact Finding: Attempting to Reverse-Engineer Factual Recall on the Neuron Level," https://www.alignmentforum.org, 22 December 2023. [Online]. Available: https://www.alignmentforum.org/posts/iGuwZTHWb6DFY3sKB/fact-finding-attempting-to-reverse-engineer-factual-recall. [Accessed 26 12 2024].

[9] M. DeBellis, N. Dutta, G. Gino and A. Balaji, "Integrating Ontologies and LLMs to Implement Retrieval Augmented Generation (RAG)," *Applied Ontology,* 2024.

[10] The Dublin Core™ Metadata Initiative (DCMI), "DCMI Metadata Terms," 10 April 2024. [Online]. Available: https://www.dublincore.org/specifications/dublin-core/dcmi-terms/. [Accessed 16 April 2024].

[11] P. Blackwood, "A Brief Introduction to the Gist Semantic Model," Semantic Arts, 3 November 2020. [Online]. Available: https://www.semanticarts.com/a-brief-introduction-to-the-gist-semantic-model/. [Accessed 19 2 2025].

[12] M. DeBellis, "A Universal Moral Grammar (UMG) Ontology," in *SEMANTiCS 2018 – 14th International Conference on Semantic Systems*, Amsterdam, 2018.

[13] L. Richardson, "Beautiful Soup Documentation," April 2007. [Online]. Available: https://www.crummy.com/software/BeautifulSoup/bs4/doc/. [Accessed 17 April 2024].

[14] A. Singhal, "Introducing the Knowledge Graph: things, not strings," Google, 16 May 2012. [Online]. Available: https://www.blog.google/products/search/introducing-knowledge-graph-things-not/. [Accessed 8 May 2023].

[15] Franz Inc., "AllegroGraph Freetext Indexing," 22 March 2023. [Online]. Available: https://franz.com/agraph/support/documentation/current/text-index.html. [Accessed 27 April 2023].

[16] D. Gulbranson, "Python Nameparser," https://github.com/, 20 September 2023. [Online]. Available: https://github.com/derek73/python-nameparser. [Accessed 19 May 2025].

[17] D. C. Anastasiu and G. Karypis, "Fast Parallel Cosine K-Nearest Neighbor Graph Construction," in *2016 6th Workshop on Irregular Applications: Architecture and Algorithms (IA3)*, Salt Lake City, UT, USA, 2016.

[18] FTSE Russell, "Industry Classification Benchmark," ICB Publications, London, 2012.

[19] R. Brulle, Interviewee, *Email discussion with Michael DeBellis.* [Interview]. 14 May 2025.

[20] Sabin Center for Climate Change Law, "RCC Ruling on Chiquita “climate neutral bananas”," Columbia Law School: Sabin Center for Climate Change Law, 2022. [Online]. Available: https://climatecasechart.com/non-us-case/rcc-ruling-on-chiquita-climate-neutral-bananas/. [Accessed 22 May 2025].