Preventing Decision Paralysis

Members

Achu Jeeju (achujeej@iu.edu)

Adam Holtzlander (adaholtz@iu.edu)

Akash Pintukumar Das (dasakas@iu.edu)

Ikramuddin Ahmed (ikahmed@iu.edu)

Sarvesh Rakesh Soni (sarvsoni@iu.edu)

Tumul Rajvedi (trajvedi@iu.edu)

Tyler Kelly (kellytd@iu.edu)

Point Of Contact

Abubaker Ahmed

Catherine Rawlek

Joy Zayatz

Sanjana Jairam

Sponsors

David Wild
Kyle Stirling

Overview

In an era of unprecedented data complexity, the human brain faces an extraordinary challenge: navigating and synthesizing vast, diverse information streams to make critical decisions. The project "Preventing Decision Paralysis" emerges as a pioneering exploration of how computing, particularly data science and artificial intelligence, can support and potentially replicate the brain's remarkable ability to integrate heterogeneous information.

Modern decision-making scenarios—spanning personal life, politics, medicine, disaster response, and competitive intelligence—demand the seamless integration of diverse data types. These range from environmental sensors and digital infospheres to structured databases, unstructured text, and complex imagery. Traditional data science methods often struggle with this multifaceted landscape, resulting in incomplete analyses, significant decision delays, or complete decision-making overwhelm.

This initiative represents a critical intervention, seeking to bridge the gap between technological capabilities and human cognitive processing. By investigating innovative technologies like knowledge graphs and advanced visualization techniques, the project aims to develop systems that not only process large-scale data but present it in ways that genuinely support human comprehension and decision-making.

Purpose / Problem Statement

Decision paralysis, also known as analysis paralysis, refers to the phenomenon where individuals or groups become overwhelmed by the sheer volume of available information, leading to an inability to make timely or effective decisions. This occurs when there are too many options to evaluate or too much data to process, ultimately resulting in delayed action or missed opportunities.

Current practices within data science often require standardized data formats, creating relatively homogenous databases, this process can often require significant time and work hours to accomplish before appropriate analysis can be conducted which can lead to delays in critical decision making at various organizational levels. There exists a need to adapt the heterogeneous landscape of data, in such a way that provides individuals with user friendly tools that allow them to harness data in a way that supports and expedites the decision making process.

The aim of this project is to attempt to build and offer structured decision-support systems/decision intelligence systems that streamline this data analysis process, reducing cognitive overload and helping individuals or organizations make quicker, more informed decisions. As a starting point, by seeking out modern technologies that can aid in the integration process and tailor these solutions to specific decision making processes, improving efficiency.

Key Challenges:

The project directly addresses decision paralysis, a phenomenon where individuals and organizations become overwhelmed by excessive information, leading to delayed or ineffective decision-making. Specifically, the research targets key challenges including:

- The vast availability of diverse datasets lacking standardization
- Insufficient toolsets designed to overcome data format heterogeneity
- The cognitive bottleneck created by expansive data volumes
- Limited visualization methods for complex information

Stakeholder Impact

This research holds transformative potential for multiple critical domains:
- *Medical Decision*-Making: Enabling more comprehensive diagnostic and treatment strategies
- *Disaster Response*: Facilitating rapid, informed decision-making in time-critical scenarios
- *Competitive Intelligence*: Providing nuanced insights across complex information landscapes
- *Policy and Governance*: Supporting more holistic, data-driven policy development.

Project Objectives

The "Preventing Decision Paralysis" project is built on a foundation of strategic objectives aimed at revolutionizing decision support systems through innovative technology and user-centered design. These objectives address the technical, cognitive, and practical challenges of integrating and navigating complex data landscapes.

1. Research and Identify Integration Technologies

    - Knowledge Graph Development: Investigate and refine techniques for building scalable and dynamic knowledge graphs capable of integrating structured and unstructured data.
    - Advanced Data Fusion: Explore methods for combining diverse data sources, focusing on harmonizing heterogeneous formats to ensure seamless integration.
    - Emerging Tools and Frameworks: Evaluate and implement state-of-the-art tools, such as large language models (LLMs), for natural language processing and data synthesis to improve automation and efficiency.

2. Architectural Design

    - Framework for Collaboration: Design a modular architecture that allows various technologies, such as graph databases, NLP tools, and visualization engines, to work together cohesively.
    - Bridging Technology and Cognition: Develop systems that emulate human cognitive processes, such as pattern recognition and relational reasoning, to facilitate decision-making in data-heavy environments.
    - Scalability and Accessibility: Focus on creating systems that are scalable for large datasets while remaining accessible to non-technical users through intuitive interfaces.

3. Visualization and Comprehension

    - Rapid Processing of Complex Data: Create advanced visualization techniques, such as interactive dashboards and graph-based visualizations, to simplify the interpretation of large-scale, multifaceted information.
    - User-Friendly Interfaces: Design interfaces that enable users to easily query, explore, and interact with the data, leveraging natural language and visual aids to reduce cognitive barriers.
    - Insight-Driven Design: Ensure visualizations highlight actionable insights rather than overwhelming users with raw data, supporting more informed decision-making.

4. Applied Research and Exploration

- Domain-Specific Prototypes: Develop and test prototypes in strategic areas, such as emergency response and pharmaceutical decision support, to validate the system's applicability across different use cases.
- Cross-Domain Adaptability: Investigate the potential to scale and adapt the system for broader domains, including policy development, competitive intelligence, and general knowledge exploration.
- Robust Applicability: Generate meaningful insights from both raw-unstructured data as well as more homogenous data sources and tables.
- Performance Metrics and Testing: Establish rigorous metrics for evaluating effectiveness, efficiency, and user satisfaction, iterating based on testing results.

By addressing these objectives, the project seeks to lay the groundwork for a transformative decision-support system. It not only aims to mitigate the cognitive and logistical challenges of decision paralysis but also positions itself as a pioneer in bridging the gap between human cognition and machine intelligence for actionable, real-world insights.

Methodology

I. Test Case Exploration : Dual-Domain Investigation

Our research focused on two critical domains to validate the decision support system's versatility and effectiveness:

A. Emergency Response Informatics

This test case seeks to develop a prototype pipeline that uses AI to automate the process of identifying and collecting relevant, high-quality data sources for emergency responders. Emergency situations require rapid decision-making based on accurate, reliable information, and the current data gathering process can be time-consuming. Our goal is to create a tool that streamlines the search and exploration phase by centralizing relevant data sources, enabling responders to quickly assess the most valuable information.

Key Deliverables

1. A curated set of data sources for a specific emergency scenario.
2. A prototype knowledge graph that ranks the information value of these sources.
3. A functional pipeline that centralizes data collection and integrates with briefing software.
4. A roadmap identifying key areas for AI-driven automation in future iterations of the pipeline.

Initially, the project focuses on a single problem area to validate the approach and work through the process manually. The final deliverable will include a detailed roadmap outlining areas of the process where AI automation can enhance efficiency and accuracy.

B. Pharmaceutical Decision Support

This test case aims to design and develop a prototype decision support solution for patients, with a particular emphasis on individuals with chronic conditions. The solution will provide users with accurate, accessible, and concise information on medications, reducing information overload for those without experience in medicine. By allowing individuals to search for specific drugs, the tool will generate a comprehensive briefing that can include key information such as:

- Drug name
- Purpose of the medication
- Active effect and expected benefits
- Active ingredient
- Side effects
- Potential drug interactions
- Dosage and administration guidelines (if applicable)
- Contraindications and warnings

The solution will serve as a decision aid for patients, empowering them with essential knowledge to make informed decisions about their treatments. The system will start with a curated set of drugs and expand over time to include a broader range of medications.

Key Deliverables

1. *Prototype Version 1*: A functional prototype able to retrieve drug information for a select number of drugs, demonstrating the tool's capabilities.
2. *Data Source Mapping*: Identification and integration of key data sources, both static and live, to feed the decision support tool.
3. *Usability Testing*: Initial testing of the prototype with a user group, potentially involving patients with chronic conditions, to evaluate effectiveness, accuracy, and user experience.

II. Data Sources

A. Emergency Response Data Sources

- Primary Data Repositories:
  - FEMA Open Data Portal
    - Disaster Declaration Summaries v2
    - IPAWS Archived Alerts v1
  - National Oceanic and Atmospheric Administration (NOAA)
  - United Nations Office for Disaster Risk Reduction (UNDRR)

- Data Collection Methods:
  - API-based retrieval OpenFEMA API DocumentationLink
  - Web scraping techniques
  - Public dataset downloads

B. Pharmaceutical Data Sources

- Primary Information Repositories:
  - FDA Drug Databases
  - World Health Organization (WHO) Medication Lists

    - ClinicalTrials.gov
    - PubMed Central

- Data Collection Methods:
    - Official API access
    - Structured database downloads
    - Validated pharmaceutical references

III. Data Formats and Processing

A. Emergency Response Data Sources

- Standardized Formats:
    - CSV (Comma-Separated Values)
    - JSON (JavaScript Object Notation)
    - XML (Extensible Markup Language)

- Processing Techniques:
    - Data normalization
    - Cross-reference validation

B. Pharmaceutical Data Sources

- Standardized Formats:
    - CSV (Comma-Separated Values)
    - JSON (JavaScript Object Notation)
    - XML (Extensible Markup Language)

- Processing Techniques:
    - Data normalization
    - Cross-reference validation

IV. Technology Exploration and Evaluation

A. Comparative Analysis: Neo4j vs Nebula Graph

**For a full report, see the associated document in Basecamp ("Neo4j vs Nebula Graph)

Recommendation:

- Choose **Neo4j** if you prioritize ease of use, mature tooling, and comprehensive documentation for medium-sized knowledge graphs.
- Opt for **Nebula Graph** if you need to handle massive datasets across distributed systems with high performance and scalability.

Neo4j

Neo4j offers robust integration with major cloud platforms, including **Azure**, **AWS**, and **Google Cloud Storage**. AWS and Google Cloud provide flexible pricing models based on **gibibyte-hour consumption**, making them cost-effective options for small-scale projects, such as prototyping. This flexibility allows for dynamic resource allocation, where storage can be monitored and scaled as needed. Utilizing cloud-based storage enables seamless access to knowledge graphs (KGs) across multiple devices, eliminating the constraint of local hosting.

Nebula Graph

Nebula Graph similarly integrates with cloud storage solutions from Azure, AWS, and Google Cloud. Its cloud integration appears to be more directly aligned with its architecture compared to Neo4j. However, given the comparisons highlighted earlier—particularly regarding tooling, extensions, and ease of use—**Neo4j is recommended** for prototyping and small-scale knowledge graph development. Neo4j's more extensive ecosystem and active tool support make it better suited for generating and querying knowledge graphs in the context of smaller projects.

Natural Language Querying

Both Neo4j and Nebula Graph can support natural language (NL) query conversion into graph queries. Using tools like **NeoDash**, it is possible to implement NL-to-Cypher conversions seamlessly, provided the knowledge graph is accessible within the NeoDash environment. Nebula Graph, while compatible with Cypher-based queries, also supports **SQL-like query syntax**, which could offer additional flexibility depending on the project's specific needs.

Recommendation and Summary:

- Choose **Neo4j** if you prioritize ease of use, mature tooling, and comprehensive documentation for medium-sized knowledge graphs.
- Opt for **Nebula Graph** if you need to handle massive datasets across distributed systems with high performance and scalability.

For the prototyping phase and smaller-scale development, **Neo4j** offers a more polished experience with its broader toolset and cloud storage integrations, making it the preferred choice. Nebula Graph, though powerful for large-scale distributed setups, may be better suited for future phases requiring massive scalability and distributed querying capabilities.

B. Technologies Considered and Evaluated

Implemented Technologies

Criteria	Neo4j	Nebula Graph
Architecture	Single-instance	Distributed
Scalability	Vertical	Horizontal
Query Language	Cypher	nGQL(Cypher Inspired)
Performance	Optimal for smaller datasets	Optimized for large-scale graphs
Community Support	Extensive	Emerging
Complexity	Lower learning curve	More complex implementation

- Graph Databases
- LLM
- Knowledge graph construction frameworks

Technologies Discussed but Not Implemented

Unexplored Technical Avenues

V. Knowledge Graph Architecture

A. Node Types and Relationships

Entity Nodes

A. Emergency Response Context
- Resource allocation
- Situational assessment
- Temporal geographical data

B. Pharmaceutical Context
- Medication entities
- Drug interaction mappings
- Patient health profiles

Relationship Mapping

Knowledge Graph Archive

Placeholder for Archive Link: [Detailed Knowledge Graph Repository Information]

Implementation Insights

Key Challenges Addressed

- Semantic variation across data sources
- Performance optimization
- Meaningful relationship mapping
- Cross-domain information integration

Conclusion

Our methodology demonstrates a comprehensive approach to addressing decision paralysis by:

- Exploring two distinct domains
- Implementing advanced knowledge graph technologies
- Developing flexible data integration strategies
- Creating a foundation for AI-driven decision support systems

The research provides a robust framework for transforming complex, heterogeneous information into actionable insights across critical domains.

Impact and Outcomes

Key Achievements

Pharmaceutical Knowledge Graph:

In the process of creating a pharmaceutical knowledge graph focused on Parkinson's disease treatments, two main approaches were utilized: the REBEL (Relation Extraction By End-to-end Language generation) model and SpaCy for Subject-Verb-Object (SVO) splits. Both methods aimed to extract structured information from unstructured text, specifically from the "Treating PD" section of the NINDS webpage.

REBEL Model Approach

The REBEL model was employed to generate triplets (subject, predicate, object) from the text. This approach showed promise in extracting meaningful relationships, such as:

- Identifying treatment relationships: "levodopa treats Parkinson's disease"
- Classifying drugs: "apomorphine has role Dopamine agonist"
- Linking drugs to symptoms: "rasagiline treats motor symptoms of PD"

However, the REBEL model also produced some inaccurate or nonsensical triplets, highlighting the challenges of automated relation extraction from complex medical text.

SpaCy SVO Splits

The second approach utilized SpaCy, a popular natural language processing library, to break down sentences into Subject-Verb-Object structures. This method aimed to capture the basic semantic structure of sentences and extract relevant information. Some useful outputs included:

- Identifying drug effects: "Levodopa reduce symptoms"
- Capturing drug interactions: "Selegiline delay need"

However, this approach also faced limitations, often producing overly generic or incorrect extractions due to the complexity of medical language and the nuanced relationships between drugs, symptoms, and treatments.

Chunking Method

A third method was attempted to break down the content scraped from the websites into multiple chunks and then feed it to the REBEL model, but it failed to produce any meaningful relations.

Measurable Outcomes

Performance Metrics

Stakeholder Impact

A. Emergency Response Domain

- Enhanced rapid decision-making capabilities
- Streamlined information gathering processes
- Improved resource allocation strategies

B. Pharmaceutical Information Domain

- Empowered patient decision-making
- Simplified complex medication information
- Increased transparency in drug information access

Technical Results

Technology Stack

- Graph Database Integration
  - Neo4j and Nebula Graph exploration

Technical Challenges - Pharmaceutical:

The technical challenges we faced while creating a pharmaceutical knowledge graph from unstructured text were:

1. Complexity of Medical Language: The specialized and nuanced nature of medical terminology in the context of Parkinson's disease and its treatments, posed a significant challenge for automated extraction methods. Both the REBEL model and SpaCy struggled with accurately interpreting complex medical concepts and relationships.
2. Ambiguity in Relationship Extraction: Determining the correct relationships between entities (e.g., drugs, symptoms, side effects) proved challenging. For example, distinguishing between a drug treating a symptom versus causing it as a side effect required nuanced understanding that current models often lack.

3. Handling Nested and Complex Sentences: Medical texts often contain long, complex sentences with multiple clauses and nested information. This structure made it difficult for both the REBEL model and SpaCy to accurately parse and extract relationships without losing context or misattributing information.
4. Inconsistent Performance of Models: The REBEL model, while effective in some cases, produced inconsistent results, sometimes generating accurate triplets and other times producing nonsensical or incorrect relationships. This inconsistency made it challenging to rely on the model's output without extensive manual verification.
5. Overgeneralization in SVO Splits: The SpaCy SVO approach, while simpler, often led to overgeneralized or oversimplified extractions that failed to capture the nuanced relationships present in the pharmaceutical context.
6. Handling of Negations and Conditionals: Medical texts often include important negations (e.g., "does not cause") or conditional statements (e.g., "may lead to"). Both methods struggled to accurately capture these nuances, potentially leading to misinterpretations of drug effects or relationships.
7. Scale and Efficiency: Processing large volumes of text efficiently while maintaining accuracy posed a significant challenge, especially when dealing with complex models like REBEL.
8. Validation and Accuracy Assessment: Determining the accuracy of extracted relationships without extensive domain knowledge or manual verification by medical experts proved to be a substantial challenge, making it difficult to refine and improve the extraction methods.

These technical challenges highlight the complexity of automating knowledge extraction in the pharmaceutical domain and highlight the need for more sophisticated, domain-specific approaches to natural language processing and information extraction in medical contexts. We recommend to use corpus to identify the medical terms better in order to extract the relations and entities.

Disaster Response Knowledge Graph:

The disaster dataset is modeled as a graph in Neo4j, capturing entities like states, disasters, incident types, areas, dates, and declaration types as nodes, with relationships representing their connections. Each state is represented as a node with attributes like its abbreviation, while disasters are central nodes linked to other entities with attributes such as their unique number, title, and declaration date. Incident types categorize disasters (e.g., “Flood” or “Tornado”), and areas represent the geographical regions impacted, such as “Statewide” or specific cities. Dates mark the timeline of events, tracking when disasters started and ended, and declaration types (e.g., “DR” for Disaster Relief) distinguish the nature of official responses.

Relationships enrich the graph by connecting these entities meaningfully. States declare disasters through the [:DECLARED] relationship, while [:HAS_DISASTER] directly links states to disasters that occurred. Disasters are tied to their types with [:TYPE] and to their impacted areas via [:IMPACTED]. Temporal relationships like [:STARTED_ON] and [:ENDED_ON] link disasters to their start and end dates, enabling duration analysis. Additionally, [:DECLARATION_TYPE] links disasters to their declaration types, and [:INCIDENT_OCCURRED] connects states to incident types, facilitating queries on frequent disaster types within a state.

This structure supports diverse analyses, such as identifying the most disaster-prone areas, understanding temporal trends, analyzing state-specific impacts, and correlating disaster types with responses. The graph's design enables deep insights into the dataset by leveraging relationships for contextualized exploration of disaster events.

Disaster Response Technical Challenges:

The process of building and querying the disaster graph database involved several technical challenges. Data modeling required precise decisions on representing entities like State, Disaster, IncidentType, and Date, ensuring the structure was both intuitive and effective for analysis. Temporal data, such as start and end dates of disasters, introduced complexity in accurately modeling time-based relationships like [:STARTED_ON] and [:ENDED_ON]. Some relationships, including [:DECLARED] and [:HAS_DISASTER], overlapped in functionality, necessitating a clear definition of their roles to avoid redundancy. Handling geographical data was challenging due to ambiguities like "Statewide" areas versus specific locations, which affected the granularity of impact analysis. Additionally, managing data consistency and cleaning issues, such as inconsistent naming conventions, required careful preprocessing. Query formulation posed another hurdle, as complex analyses—such as identifying shared disaster types or calculating average durations—demanded optimized Cypher queries to ensure performance. Balancing the use of attributes versus relationships further added to the complexity, requiring trade-offs between query efficiency and graph flexibility. Despite these challenges, the final graph model supports robust analyses of disaster trends, impacts, and responses, enabling efficient and meaningful insights.

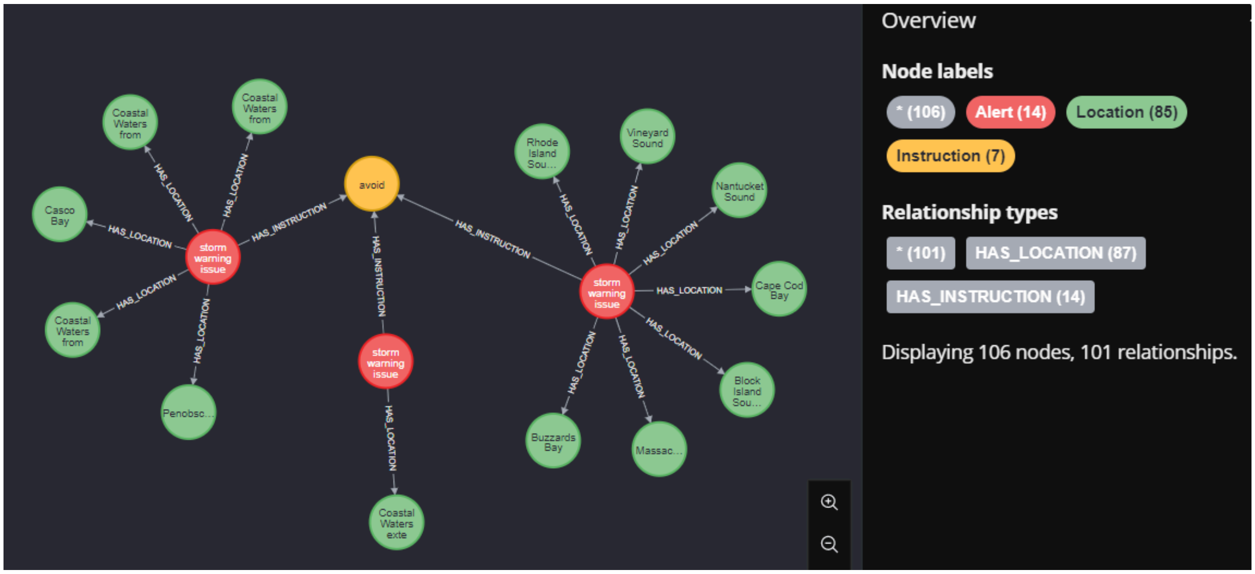
Emergency Alert Knowledge Graph:

This knowledge graph, derived from the Integrated Public Warning and Alert System dataset, comprises over 275,000 nodes and 1 million relationships, encompassing a dataset of over 250,000 alerts. The graph centers on alert-specific information, including event type, severity, location, and recommended actions. Key node types include Alerts, Locations, and Instructions, with relationships linking Alerts to Locations using [:HAS_LOCATION] relation (indicating impacted areas) and Instructions using [:HAS_INSTRUCTION] relation (providing guidance). This rich dataset allows for exploration of historical alerts based on location, event type, or specific instructions, aiding in understanding past responses and potentially informing future decision-making during emergencies. This knowledge graph has the potential to be a valuable tool in disaster response and emergency management, particularly for decision paralysis research.

Emergency Alert Technical Challenges:

The primary technical challenge encountered during the development of this knowledge graph was the immense size of the IPAWS Archive Alert Dataset. This dataset far exceeded the capacity of standard computing environments, such as Google Colab or personal PCs. To mitigate this issue, the code was optimized to efficiently handle smaller data subsets. Initially, Neo4j Aura DB was considered for graph development and analysis. However, the free tier limitations imposed by this platform necessitated a shift to Neo4j Desktop. This change allowed for the successful development of the graph, despite the reduced capabilities. The sheer volume of data within the dataset presented another significant challenge. The processing and loading of over 275,000 nodes and 1 million relationships required substantial computational resources and time. As a result, the execution time for the entire process extended to approximately 1.5 hours. To facilitate efficient sharing and analysis, a database dump was generated, enabling direct access to the graph without the need for repeated processing.

Emergency Alert Graph:



Overall Technical Challenges:

The overall technical challenges related to the process of translating natural language into a cypher query involved a lack in API understanding and education around using the OpenAI Library to query ChatGPT and subsequently return a Cypher query estimate based on information given in a prompt. As a stand alone proof of concept, this process works within the native ChatGPT environment, but the planned integration tools are not yet implemented to a capacity that supports the entire data pipeline from user input to knowledge graph to then the ultimate reporting brief as an endpoint.

The issues surrounding API issues are mirrored in the Neo4j environment where the Text2Cypher extension for NeoDash returns 429 errors for exceeding requests. Our team has experimented with multiple API keys and continue to get this error. Possible workarounds include testing different API key subscriptions specifically for ChatGPT or other OpenAI products.

Implementation Constraints:

Constraints involving the implementation of our design system as a deliverable product mostly are about the aforementioned technical challenges. The design of our pipeline relies on the ability for multiple different interacting technologies to cooperate. Our team initially tackled this problem by creating Knowledge Graphs from structured and unstructured data, and then attempting to query these graphs using natural language. A major barrier here was the hardware limitations to storage of the Knowledge Graphs, accurate tokenization of unstructured data and implementations of required APIs. These constraints limit our ability to arrive at a complete solution, however with more education can be achieved.

Project Conclusion:

The "Preventing Decision Paralysis" project has successfully transitioned from conceptualization to the early stages of implementation, marking a significant milestone in addressing the complexities of decision-making in data-rich environments. While the current state of the project involves functional prototypes, there is a clear path forward for further development.

Key Achievements to Date

1. **Natural Language Information Retrieval**:
   Initial prototypes demonstrate success in extracting and processing information from natural language inputs. This represents progress in bridging the gap between unstructured data sources and actionable insights.
2. **Structured Data Knowledge Graphs**:
   Building knowledge graphs from structured data sources has been more successful, allowing us to demonstrate functional frameworks and preliminary querying capabilities.
3. **Querying with Text2Cypher**:
   Early experiments with tools like Text2Cypher show promise in converting natural language queries into Cypher searches, streamlining the user interface for interacting with knowledge graphs.

Key Challenges and Hurdles

1. **Relational Information from Unstructured Data**:
   Extracting and converting relational information from unstructured data into a working knowledge graph remains a significant challenge. Current efforts focus on either improving existing relation extraction models or identifying more suitable tools for this purpose.
2. **Knowledge Graph Storage and Accessibility**:
   Knowledge graphs are currently stored locally, which limits collaboration and scalability. A key objective moving forward is to optimize storage solutions, potentially transitioning to a centralized or distributed system to enable shared access and enhance usability.

Problem Solution Validation:

Currently the project sits at having some prototypes with varying degrees of success. We are currently still a ways away from the endpoint deliverable solution hinted at in our problem statement, and so future work needs to be done (see below) in order to further develop a solution that can be implemented.

Key Success Indicators:

- Semi-Successful Natural language querying:
  - The current solutions for natural language querying seemingly work on some toy examples and serve as proof of concept of the use case of LLMs for making cypher queries.
- Prototypical Knowledge Graphs
  - We have successfully generated some prototypical knowledge graphs which can be used in the future for further improving on query generation/search, as well as serving as a benchmark for attempts at automated generation.
- Data validation (for test cases):
  - We have generated knowledge graphs that can serve as benchmarks for future work, and in the process of doing so we also have identified and validated useful sources of data for the domains explored
- Technology identification/implementation
  - Through this process we have identified relevant and useful technologies (text2cypher, NebulaGraph, REBEL, SPACEY and others) that will be useful for further developing this project,

Recommendations for Future Work

- **Improved Relation Extraction**: Developing or implementing a more robust model to pull relational data effectively from unstructured natural language sources. The current models employed are limiting efficacy in the use of non structured data, meaning implementation is limited to topics with such sources available
- **Streamlining Data Pre-Processing** Currently, while we have been able to generate prototype knowledge graphs from structured data sources, generation of these requires significant pre-processing of the data, which is both time-consuming and relatively source specific. In order to create a tool that is usable by those not domain experts, the pre-processing step must become more streamlined for relevant data sources.
- **Optimized Storage Solutions**: Transitioning knowledge graph storage to a scalable infrastructure to allow multi-user access and integration with other tools.
- **Enhanced Querying Capabilities**: Refining natural language querying tools to improve accuracy and efficiency in generating structured queries for knowledge graph interaction. Currently we are using text2cypher, however this is limited in its applicability as it is specifically a neo4j

extension. If moving forward there is desire to use a different language, there will need to be a more general use method that can apply to different software..

Long-Term Vision / Future Predictions:

The long-term vision of the "Preventing Decision Paralysis" project centers on developing a transformative decision-support tool that seamlessly integrates advanced technologies to aid in navigating complex, data-rich environments. At its most advanced stage, this tool will enable users to pose domain-specific questions (e.g., "What do I need to know about Parkinson's?") and receive actionable insights through an automated, comprehensive process.

Future Functionalities and Goals

1. **Dynamic Data Gathering**
    - The system will autonomously identify and collect relevant data sources, both structured and unstructured, tailored to the specified question or domain space.
    - Integration of live data feeds and repositories to ensure up-to-date and contextually relevant information.
2. **Knowledge Graph Generation**
    - Automatically create a robust knowledge graph from the gathered data, combining structured and unstructured inputs into a unified representation.
    - Employ cutting-edge models for relation extraction to improve accuracy and completeness, particularly for unstructured text.
3. **Natural Language Querying**
    - Enable intuitive interactions with the knowledge graph through natural language queries, minimizing the need for technical expertise.
    - Further refine tools like Text2Cypher to enhance usability and response precision.

Predicted Impact Across Domains

- **Medical Research and Decision-Making**: Facilitate researchers and practitioners in synthesizing complex datasets for comprehensive disease understanding and treatment strategies.
- **Emergency Response**: Enable responders to rapidly gather, visualize, and act on critical situational data.
- **Policy Development**: Provide policymakers with nuanced, data-driven insights across diverse information landscapes.
- **General Knowledge Exploration**: Offer individuals an accessible tool for deep, informed exploration of complex topics, democratizing access to decision-support systems.