Data Science Program

Capstone Report - Spring 2025

Improved Multi-Agent Knowledge Sharing System using Knowledge Graphs for News Bias Detection and Fact-Checking

Christopher Washer,
Modupeola Fagbenro,
Pavani Chella

supervised by
Amir Jafari

Acknowledgements

We would like to express our sincere gratitude to our advisor, Amir Jafari, for their invaluable guidance, encouragement, and support throughout the course of this research. His insightful feedback greatly enhanced the quality of this work.

We also extend our thanks to the faculty and staff of the **Department of Data Science at The George Washington University** for providing an excellent research environment.

Abstract

This study introduces a novel approach to exploring how multi-agent system with knowledge graphs framework enhances news bias detection and fact-checking. Automated fact-checking and verification systems have evolved significantly, often struggling with complex contextual information and advanced narratives. This project hypothesizes that integrating traditional knowledge graphs with multi-agent architectures can significantly improve performance in identifying bias and verifying facts in news article content. This research investigates whether there is a statistically significant improvement in accuracy, precision, and recall when knowledge graphs are incorporated into multi-agent systems compared to systems without dynamic knowledge graph integration. The system was designed and implemented as a framework where multiple specialized agents using large language models (LLMs) collaborate while building and referencing a dynamic knowledge graph that evolves as new information is processed. Experiments conducted on diverse political news datasets show that the integration of a shared knowledge graph improved overall performance, with notable enhancements in analyzing and detecting bias patterns and contextual inconsistencies. These research findings suggest that integration of knowledge graph representation through knowledge graphs provides critical grounding to multi-agent systems, enabling more robust reasoning about media bias and factual accuracy in complex news environments.

Contents

1 Introduction

1.1 Background

1.1.1 Misinformation and Media Bias In The News Landscape

Misinformation and bias in the news media are widespread issues in today's digital media landscape, with real societal implications. Misinformation refers to false, misleading, or inaccurate information spread (intentional or not), proliferating rapidly on social media. The ease of producing and sharing content online leads to false information reaching millions, which makes it a significant threat to public interests [1]. Filling social media feeds with misleading claims can confuse and overwhelm the public and erode trust in institutions and elections [2]. Examples such as false stories about the 2020 U.S. presidential election suggest that misinformation can impact political outcomes. The issue is well known enough that over half of Americans expect political misinformation to worsen in their lifetimes, highlighting the increasing prevalence of this problem [3][4]. Media bias refers to partiality in news reporting through tone, framing, or story selection that skews information in favor of a particular viewpoint. News media bias often results in audiences self-selecting news sources that support their existing beliefs, and the sources can reinforce them by presenting information through a partisan lens. Over time, biased presentation and consumption can produce different versions of reality for different audiences. A recent study of Breitbart, a conservative news outlet, and the New York Times, a left-leaning news outlet, found that each outlet was relatively neutral in coverage, but exhibited a polarizing attitude towards specific entities, contributing to disparities in audience political views [5]. This finding supports similar work looking at these two news outlets, showing how changes to similar common words can impart a different meaning to their audiences [6]. Therefore, even when mainstream outlets aim for accuracy, selective emphasis on certain topics or certain word choices can reinforce political polarization. Continued research is needed to counter these trends in media bias and misinformation to improve the political news landscape.

1.1.2 Knowledge graphs in information processing

Knowledge graphs (KGs) are a structured way of representing and organizing complex information using a network of nodes (entities such as people, organizations, events, or concepts) and edges (relationships between those entities), typically expressed as subject-predicate-object triplets [7]. This structure allows information to be stored in a way that mirrors real-world relationships.: For example, a KG can represent the fact that a politician belongs to a specific political party with the triplet:' Politician - memberOf - party'. These triplets allow for context-rich storing of information as the connections between nodes model the logic and structure of content. Linking discrete data points into a graph enables structure-based and semantic reasoning by traversing nodes and connections to infer new insight or answer complex queries [8]. Unlike unstructured text or relational databases, the interconnected structure of a KG supports advanced retrieval and inference.

KGs are integral to many complex information systems that require access to structured knowledge [9]. Search engines use massive KGs (such as Google Knowledge Graph) to enrich search results. In the context of news and fact-checking, KGs offer a way to compile and link vast amounts of information about entities, claims, and sources. A news oriented KG can connect a claim to who said it and what evidence supports or refutes it, enabling automated reasoning about truthfulness [10]. For bias detection, a KG can represent a network of articles with source or political bias information and help identify patterns such as grouping articles that frame an event similarly. A recent study built a political KG and showed that it improved political perspective detection in news media [11]. The KG provided structured context from real-world political dynamics that are missing in similar text-only models, highlighting how KGs can aid processing tasks by providing broader factual and relational content.

1.2 Problem Statement

1.2.1 Challenges in news evaluation

Evaluating news for bias and misinformation is a difficult task, with human and computational limitations. One major challenge is the daily number of news articles and social media claims that emerge outpaces what human fact-check teams or editors can feasibly review. Studies find that professional fact-checkers may spend hours on a given claim, making purely manual verification insufficient against the increasing tide of online information [12]. This leads to a verification delay; by the time a human analysis is complete and published, the misinformation may already be widely spread. Bias may often be complicated by the reader; what one person perceives as biased, another might view as reasonable, making it difficult for ground truth bias detection. Automated approaches have other difficulties. Automated bias detection is challenging because bias can be nuanced: it may appear through tone, omission of context, or choice of sources, which require understanding beyond surface text features. Early machine learning models often picked up on spurious signals; for example, a classifier could mainly learn the source publication as a proxy for bias without analyzing the content [13]. This limitation often reduces generalizability for new or moderate outlets.

Automated fact-checking has similar challenges, as natural language processing-based approaches lack world knowledge and context to spot subtle misinformation or biases. Traditional models check if phrasing or metadata resembles known false statements and might struggle without contextual knowledge. Modern large language models (LLMs) bring in-depth language understanding, but despite this methodology, they still suffer from hallucinations and produce confident-sounding but incorrect answers[14]. LLMs are constrained by what they have seen in training, so recent research is often not in their knowledge base, which is essential for news-based fact-checking.

1.2.2 Need for automated fact-checking and bias detection

Given these challenges, there is a clear need for automated approaches to assist in fact-checking and bias detection. No newsroom or fact-checking team can keep up with the influx of online claims, so scalable AI agents are needed to help monitor and evaluate content [12]. Automated systems can rapidly flag potential misinformation or extreme bias for news teams to review. While automation is essential, it is important to combine it with robust knowledge resources to improve accuracy. By connecting LLM-based agents to an external KG, we can ground their analysis in up-to-date, verified information. The LLM brings comprehension and reasoning, while the KG provides retrievable evidence that the LLM can use to make informed judgments. Overall, the high volume of content, the nuance of bias detection, and the knowledge limitations of artificial intelligence make news evaluation a difficult problem. For these reasons we suggest that KG-integrated multi-agent systems offer an improved approach to combating misinformation and media bias over traditional or LLM-only methods.

1.3 Research Objectives

1.3.1 Primary objectives

1. Assess the impact of shared KG-based memory in a multi-agent system on news bias detection and fact-checking accuracy.

2. Measure the performance difference between multi-agent systems with and without KG integration to show the effect a KG can have in addressing complex news evaluation tasks.

1.3.2 Specific research questions

1. What is the impact of knowledge graph integration into multi-agent system capabilities in bias detection, fact-checking, and overall system evaluation?

1.4 Significance of the Study

This research makes significant contributions to both theoretical understanding and practical applications in the field of media analysis through innovative integration of multi-agent systems and knowledge graphs.

By demonstrating how knowledge graphs can serve as a shared memory repository between intelligent agents, this research establishes a framework for more sophisticated collaborative systems that exceed the capabilities of isolated approaches. The integration of semantic knowledge representation with large language model capabilities creates a more robust and explainable system for complex information processing tasks.

Furthermore, this research has direct applications for journalism professionals, media organizations, educators, and everyday news consumers seeking to navigate an increasingly complex information landscape. The methodology developed provides a blueprint for creating systems that can scale to address the volume and variety of modern news content while maintaining high accuracy in detecting bias and verifying factual claims.

1.5 Scope and Limitations

This study focuses on the design, implementation, and evaluation of a multi-agent system integrating knowledge graphs for news bias detection and fact-checking The study examines news content primarily within the US political domain, focusing on articles from mainstream news outlets that cover politically sensitive topics.

The system examines and implements two primary specialized agent functions: fact-checking and bias detection, leveraging large language models augmented with knowledge graph integration.

Despite the contributions, this research acknowledges several limitations which includes the system is primarily designed and tested on US political news content, which may limit generalizability to other domains or cultural contexts without additional adaptation.

Additionally, knowledge graphs require regular updates to remain current, and the system's effectiveness may decrease for very recent events not yet represented in the knowledge base.

Furthermore, the integration of large language models with knowledge graphs creates significant computational demands that may limit real-time application in resource-constrained environments.

1.6 Dataset Description

1.6.1 Source and characteristics

The news KG integration to support this system was constructed from a comprehensive corpus of political articles collected via the NewsAPI aggregator [15]. We focused on a curated list of 20 well-known news outlets spanning the ideological spectrum, including ABC News, Associated Press, Axios, Breitbart News, CBS News, CNN, Fox News, MSNBC, National Review, NBC News, Newsweek, New York Magazine, Politico, Reuters, The American Conservative, The Hill, The Huffington Post, The Washington Post, The Washington Times, and USA Today. These sources were selected to ensure a balance of political leanings and editorial standards. Each source's known bias was characterized using ratings from AllSides [16], which provides crowd-sourced reviews augmented by expert editorial analysis. AllSides' media bias classification (on a three-point left-center-right scale) was used for the political bias of each source, and this metadata was associated with the corresponding source and article entries in the knowledge graph.

For fact checking, we compiled a benchmark data set by scraping the Daily Fact Checks of Media Bias / Fact Check (MBFC)[17]. MBFC publishes daily summaries of fact checks drawn from internationally recognized fact-checking organizations. For bias detection, a separate corpus of articles was assembled, each article labeled according to its source's AllSides bias rating. This data set reflects a mix of left, center, and right biases based on the outlet of origin, serving as ground truth for testing the bias detection capabilities of the system.

2 Related Work and Literature Review

2.0.1 Existing News Evaluation Systems

In response to increasing misinformation, platforms such as Ground News[18] and AllSides use comparative bias tracking to analyze stories across various political media outlets. These platforms give news consumers an understanding of the of the bias associated with sources and news coverage but may not address individual news articles or independent media sources.

Academic researchers have proposed solutions employing natural language processing to evaluate factual accuracy and bias[19]. While this effectively detects coverage discrepancies, they have often faced a problem operating in isolation. This research provides groundwork for automated bias detection and fact checking but does not do so in a multi-agent collaborative fashion which may limit the overall functionality in real world application.

2.0.2 Multi-Agent Architectures

Multi-agent systems have become popular in knowledge-intensive domains because of their modularity and distributed problem-solving capabilities. Multi-agent systems were created as a framework for agent autonomy in foundational research work [20]. Recent studies have shown the effectiveness of implementing large language models (LLMs) as the decision makers and operators[21][22]. These system architectures, orchestrated by platforms like LangChain[23], facilitate shared memory and modular reasoning among agents, enhancing their collaborative interaction. Despite ongoing advanced research, there is limited application of integrating these systems with a shared knowledge graph-based memory or applying them to news evaluation tasks.

2.0.3 Knowledge Graph Applications in Information Processing

KGs are fundamental in structured information representation with semantically enriched storage for entity-relation triplets. KGs often used enable the connection of diverse news topics and sources through semantic reasoning and logic. Tools like Neo4j have robust capabilities designed to make them more suitable for dynamic, evolving datasets such as political news. Recent initiatives [24][25]investigate the integration of knowledge graphs with transformer-based models to improve claim verification and event linking. However, building a KG to support a multi-agent setting in this domain, remains unexplored.

2.0.4 Fact-Checking Methodologies

Fact-checking is a popular area of research for the application of LLM-based systems. Approaches have progressed from simple binary verification to more complex, context-sensitive analysis. Recent methodologies utilize transformer-based models trained on datasets such as LIAR[26] and BABE [27], while earlier research findings in similar systems, like ClaimBuster [28], implemented rule-based filters and similarity metrics [29]. Recent advancements have explored chaining together multiple facts through multi-hop reasoning and entailment analysis to verify claims[30]. Nonetheless, there are still concerns regarding hallucinations in outputs generated by large language models, the weighting of source credibility, and scalability across various human domains and languages. The integration of a structured memory via KG into a fact-checking processes is a promising opportunity to improve on past in verification workflows.

2.0.5 Bias Detection Approaches

The identification of bias in news includes story selection, tone, or choice of sources. Linguistic models have been trained to recognize subjective words, sentiment polarity, and modality [31]. Recent efforts leverage models which account for political alignment, narrative framing, and language cues[32]. LLMs fine-tuned on news bias-related corpora have demonstrated improvement in bias detection, particularly when paired with external contextual data.

2.0.6 Research Gaps and Opportunities

There are of a growing number tools and methodologies in the domain of news evaluation but various limitations and setbacks have slowed advancement. First, effective solutions often work in isolation lacking the collaborative nature multi-agent system designs. Second, while KGs offer semantic depth, their integration with agent-based reasoning remains underexplored. This study aims to address these research gaps by (i) implementing a multi-agent system driven by a large language model, (ii) utilizing a knowledge graph (KG) as a shared memory, and (iii) evaluating the system's effectiveness through metrics such as bias detection and fact-checking accuracy in comparison to an LLM lacking KG integration. Such a system will contribute to the advancement of automated journalistic integrity, improve media transparency, and inform public conversation.

3 Initial System Design and Methodology

3.1 Initial System Architecture

The system's original approach for multi-agent systems uses dynamic knowledge graphs to detect media bias in news articles and provide fact-checking of news topics/articles (Figure 1).

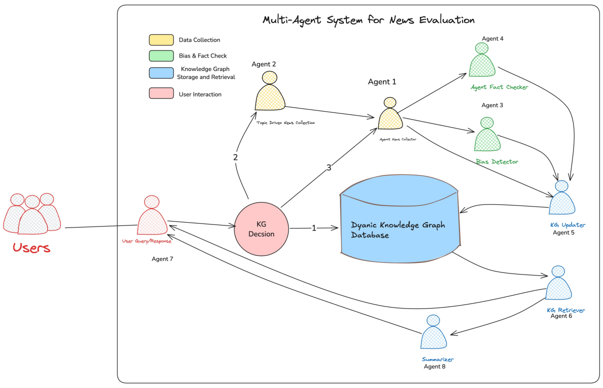


Figure 1: Original system architecture for this project

Our initial system consisted of six specialized agents that work together:
1. News Collector Agent: Collects news articles from NewsAPI and other sources
2. Bias Analyzer Agent: Analyzes news articles for potential bias and political leaning
3. Fact Checker Agent: Verifies factual claims in news articles
4. Updater Agent: Maintains the knowledge graph with new articles and analysis
5. Retriever Agent: Retrieves balanced information from the knowledge graph
6. Summarizer Agent: Agent that summaries the overall result output

3.2 Initial System Workflow

Route 1: Direct Query Processing

1. User provides a specific claim or article query
2. Fact Checker Agent checks knowledge graph for existing fact checks
3. If needed, perform new fact-checking analysis
4. Returns verified facts to the user

Route 2: News Collection and Analysis

1. News Collector Agent gathers recent political news
2. Bias Analyzer Agent evaluates articles for bias and verifies claims
3. Updater Agent integrates analyzed articles into the knowledge graph
4. Retriever Agent provides balanced information when requested

3.3 Summarizer Agent - Built But Not Implemented

3.3.1 Summarizer Agent Implementation

The Summarizer Agent focuses on producing concise, balanced, and informative summaries of news articles by intelligently integrating fact-checking outcomes and bias detection insights. Its implementation leverages both direct article content and the enhanced contextual signals provided by the Knowledge Graph (KG) and other agents' outputs. The Summarize Agent was not implemented due to time constraints.

Content Extraction: Upon receiving a request, the Summarizer Agent parses the input article or collected results (fact-checking and bias outputs) to extract the main claims, key entities, important facts, and the general narrative structure. This parsing ensures that only the most critical and relevant information is considered for summarization.

Contextual Integration: The agent integrates information from multiple sources, particularly:

- Fact-Checking Results (claims verified, evidence found, veracity scores)
- Bias Detection Results (identified bias, confidence levels, shared entities)
- Source attribution details. This combined context allows the Summarizer Agent to create a well-rounded view that incorporates veracity and neutrality insights alongside the original article content.

Prompt Construction: Based on the type of summary requested (general, fact-check focused, or bias analysis focused), the agent constructs a specialized prompt.

- For **general summaries**, the prompt emphasizes major points, contextual background, and implications.
- For **fact-check summaries**, the prompt highlights the verification results, evidence cited, and credibility evaluation.
- For **bias summaries**, the prompt stresses bias indicators, bias direction, and comparative media analysis. This prompt is structured to guide the LLM toward generating a focused, high-quality summary.

LLM Summary Generation: Using the constructed prompt, the Summarizer Agent leverages a Large Language Model (LLM) to generate the summary text. The LLM produces:

- A short, coherent summary
- A list of key points extracted
- A confidence score reflecting the model's certainty
- A list of attributed sources referenced in the reasoning

Validation and Output Structuring: The output from the LLM is validated to ensure it meets the structural expectations (e.g., fields filled, confidence score between 0 and 1). The validated result is then packaged into the SummaryResult model, ready to be returned to the user or integrated into a broader evaluation report.

This implementation uses the outputs of other agents (fact-checker and bias detector) to enhance the summarization quality. While an LLM alone can summarize text, by providing it with fact-checked claims, bias indicators, and verified sources as context, the Summarizer Agent enables more informed, balanced, and trustworthy summaries.

A central Agent Manager coordinates the Summarizer Agent alongside other components to orchestrate user queries, manage data passing, and deliver integrated insights.

3.4 Initial Implementation Plan

In the original approach, we plan to implement and carry out this research using various advanced and technological tools which includes

- Large Language Models: Claude 3 via AWS Bedrock
- Knowledge Graph: Neo4j
- Backend: Python
- API Integration: NewsAPI for article collection
- Testing Framework: Pytest

3.5 Initial Design Challenges

Credential Management: AWS session tokens in educational environments expire frequently; its required implementing robust error handling and graceful fallbacks during the unit testing. In integration tests, it is needed to skip rather than fail when credentials are unavailable.

LLM Prompt Engineering: In creating effective prompts for Claude to detect bias, multiple iterations were required; balancing prompt specificity with flexibility for diverse news content was challenging. Ensuring consistent JSON output format for inter-agent communication

State Management: Maintaining consistent state across multiple agents required careful design and backward compatibility issues when handling both singular and plural article fields. Error propagation between agents needed special attention

Performance Considerations:

- Optimizing LLM token usage to control costs
- Managing memory usage when processing large batches of articles

These challenges required thoughtful architectural decisions, robust error handling, and adaptive design patterns to create a resilient and effective system.

Design Modification Process With the challenges faced in our initial system design, we reassessed the design and integration, making it more flexible and robust. Addressing the challenges faced above , we took other measures and approaches to the overall system design and the architecture, simplifying it down to a more flexible approach while maintaining the initial function and operationality. We reduced the agent interaction and workflow processing. Allow the new architecture to interact directly more with the KG rather than interacting with themselves, making the system more simplified and more accurate

3.6 Final System Architecture and Workflow

3.6.1 System workflow Design

The system has three main paths in order to provide robust capability in news evaluation. Figure 1 shows the The main path consists of fetching news from the NewsAPI and building the KG. The user interacts with the chatbot, and the agent manager directs queries to the most appropriate agent or the KG itself. Each agent operates independently and directly with the KG.

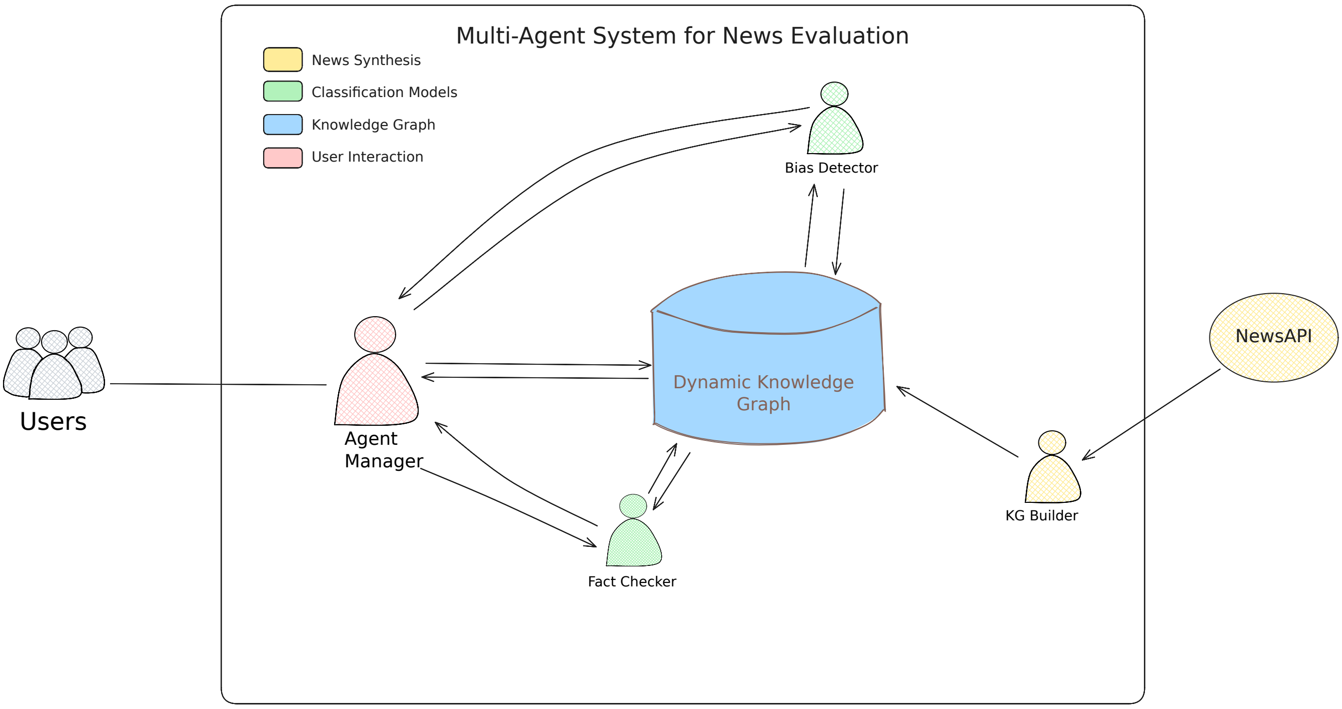


Figure 2: The system workflow architecture

The next two paths are specific to the Fact Checker and Bias Detector. They independently receive queries and interact directly with the KG to provide a response. The decision-making workflow is shown in Algorithm 1.

Algorithm 1: System decision making workflow for processing user queries.

```
Input: User request (news search, bias detection, or fact-checking)
Output: Processed output based on selected path
Main Workflow Paths
    1. KG Builder fetches news from NewsAPI and updates the Knowledge Graph;
    2. Bias Detector and Fact Checker interact directly with the Knowledge Graph;
    3. Agent Manager handles user interactions and can trigger any of the above
       processes;
    4. Each Agent operates independently, interacting via the Knowledge Graph;
Fact Check Path
    1. Route directly to the Fact Checker Agent;
    2. Triggered when user provides a specific claim for verification;
    3. Bypasses News Collection and Bias Analysis;
Bias Analysis Path
    1. Route directly to the Bias Analyzer Agent;
    2. Triggered when user wants to analyze article bias;
    3. Bypasses News Collection;
return Output generated by selected agent(s) via Knowledge Graph
```

This architecture facilitates flexibility as components do not directly rely on one another, but each agent collaborates independently via the shared knowledge graph.

3.6.2 Agent Manager Design

The agent manager controls the operation and overall workflow in the architecture effectively. The manger coordinates to which agent, if any, user queries should be sent and ensures the response is returned back to the user.

Algorithm 2: Agent Manager Path Determination

```
Input: User input query
Output: Selected processing path
Agent Manager Workflow
    1. Receive user input;
    2. Analyze input to determine intent;
    if input contains a claim then
        FactCheckPath();
    else if input is bias query then
        BiasPath();
    else
        FullWorkflow();
    3. Route request to the appropriate agent(s);
    4. Return results to the user;
FactCheckPath: Fact Check Path
    Direct input to Fact Checker Agent;
    Skip news fetching and bias detection;
BiasPath: Bias Analysis Path
    Direct input to Bias Detection Agent;
    Skip news fetching and fact checking;
FullWorkflow: Full Path
    Trigger news fetching via KG Builder;
    Then run Bias Detection and Fact Checking in sequence;
```

3.7 Fact Checker Agent Design and Implementation

The agent fact-checker is designed to verify the truthfulness of user-provided claims by using the shared KG and LLM—Claude 3.5 Sonnet's—reasoning abilities. Its implementation follows a pipeline that combines structured retrieval and natural language analysis.

1. Claim parsing: Upon submission of a claim for verification by a user. fact-checker initially analyzes the claim utilizing LangChain's graph transformer to discern its essential components and produce triplets.

2. Query KG: The agent then queries the KG using a Cypher query template and the extracted triplets to find similar facts or context. Any information that supports or refutes the claim is retrieved.

3. Prompt Construction: The fact checker builds a prompt for the LLM that incorporates the original claim and the supporting evidence from the graph. In the prompt, LLM is

directed to prioritize the information provided by the KG over its internal information to weigh the facts from the KG.

4. LLM Analysis and Verdict: The LLM processes the prompt and weighs the claim against the evidence. The fact checker determines whether the claim contradicts the evidence or aligns with it and produces a true/false response and provides the reasoning for its determination

The fact-checker agent effectively operates as the interface between the structured information in the KG and the unstructured reasoning of the LLM. In its current implementation, the KG is the primary source of evidence for the agent. By design, this agent can also be extensible to search the web for information that is not in the KG, but this was not included in this initial work.

3.8 Bias Detection Agent Implementation

The Bias Detection Agent focuses on determining the presence and lean of political bias in a provided news article by comparing it with other coverage of the same subject in the KG. Its implementation uses the KG as a basis for contextual comparisons, using other articles as a baseline to judge the slant of the target article.

1. Article Parsing: Upon receiving a news article to analyze, the agent leverages LangChain's Graph Transformer to extract the key topics, entities, and relationships in the article and produce triplets.

2. Bias Retrieval from KG: The agent queries the KG for the most structurally similar article in the KG, meaning the identified article shares the most nodes and edges as the provided article. The agent retrieves only the bias of the most structurally similar article and passes that to the LLM.

3. Bias Prompt Construction: The Bias Detection Agent assembles a prompt using the provided article contents, the bias of the most similar KG article, and the shared entities between the articles. The LLM is prompted to use the evidence provided by the similar article as a clue but also to evaluate the article for specific tone or word choices that may indicate a bias.

4. LLM Bias Classification: The LLM analyzes the prompt, provides a "Left," "Center," or "Right" verdict, and offers good reasoning for this determination capability.

An article by itself may be difficult to determine the bias, but by providing structurally similar bias in articles and the shared entities as a clue, the LLM can make a better determination of the provided article's. The KG provides key context that an LLM alone will likely miss.

3.9 Knowledge Graph Components

In this section, we describe how the KG is built, its schema, and how agents interact with it so it can be the central shared memory in the multi-agent system. Figure 2 shows a portion of an example KG that uses matching nodes and relationships as the one used for this effort.

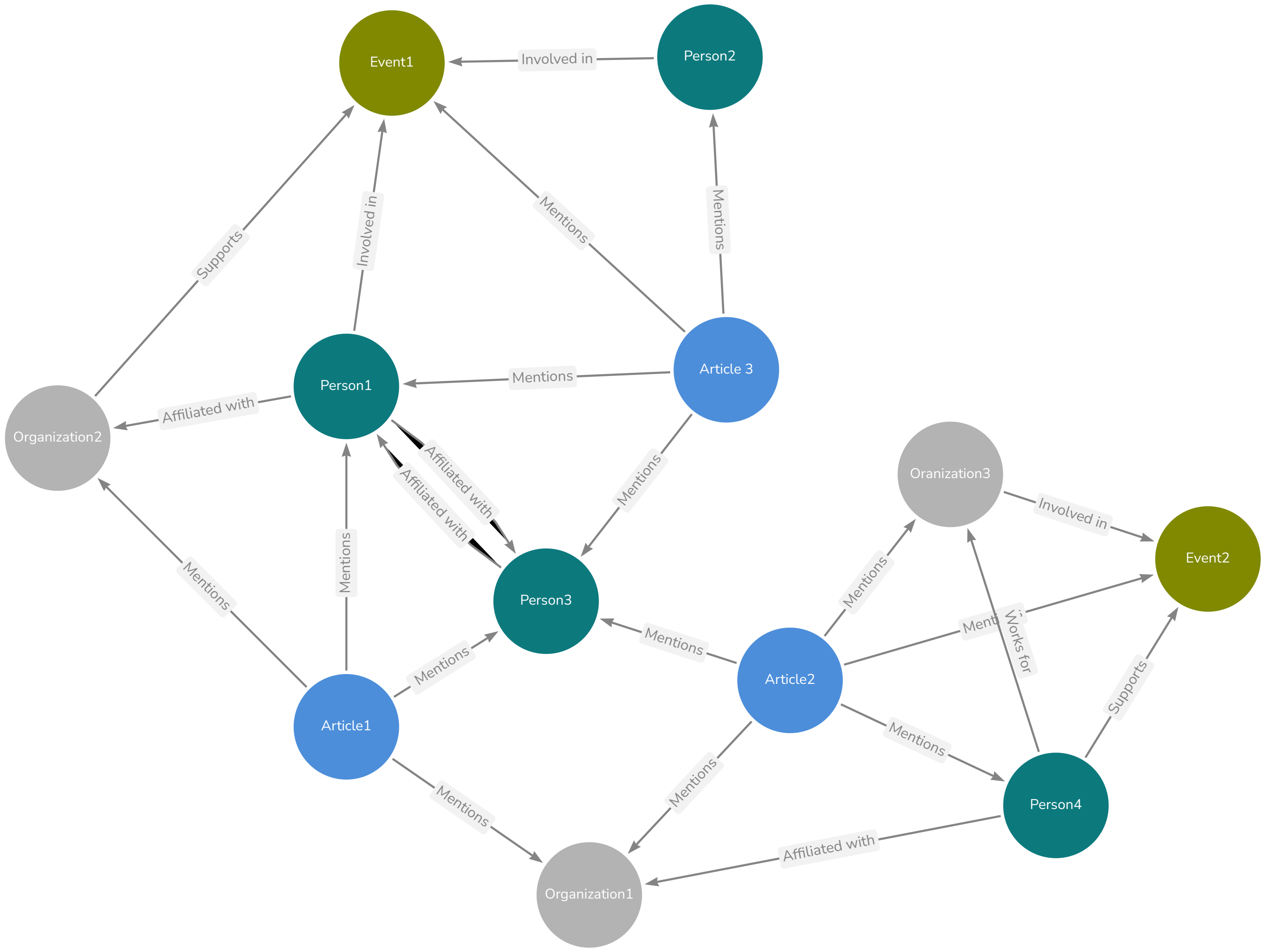


Figure 3: An example portion of the knowledge graph schema. Blue nodes represent news articles in the corpus from which the other entity nodes have been extracted. Several articles can be related to similar entities enabling complex relationships between the nodes.

3.9.1 KG Builder

The KG Builder module constructs and updates the knowledge graph using raw textual inputs. The KG Builder utilizes LangChain's LLM Graph Transformer and Claude 3.5 Sonnet to convert unstructured news article text collected from the NewsAPI into structured graph data (Figure 2). The builder identifies entities and relationships from news articles and structures them into triplets that can be added to the KG. The KG builder is configured to allow certain nodes and relationships relevant to U.S. political news to ensure that the KG stays relevant to the scope of this effort, maintains a structured format, and reduces noise in the content. The builder functions also include methods for avoiding duplicate nodes when the same entity appears. The Graph Transformer also creates additional relationships in the KG that are not specified in the schema. This results in a robust, structured, and interconnected knowledge base for the news articles that our agents can access to address user queries.

3.9.2 Knowledge Representation Schema

The schema of the news KG is designed to capture the important elements of the news content and their interconnections. The KG is centered around news article nodes that connect to the other types of nodes in the KG. This approach allows us to make queries traceable to news articles and better understand complex relationships in the data. We define several types of nodes and edges:

Algorithm 3: Initialize News Knowledge Graph Schema

```
Input: News article corpus
Output: Structured Knowledge Graph (KG)
Define Nodes
    Article;
    Person;
    Organization;
    Event;
    Policy;
    Issue;
    Location;
    Election;
    Bill;
    Vote;
    Speech;
    Alliance;
Define Edge Types
    Mentions (Article → Any Entity);
    Affiliated With (Any Entity ↔ Any Entity);
    Participated In (Person → Event);
    Endorsed (Person → Policy);
    Sponsored (Person → Bill);
    Gave Speech (Person → Speech);
    Involved In (Person → Event);
    Organized (Organization → Event);
    Supports (Organization → Policy);
    Lobbied For (Organization → Bill);
    Takes Place In (Event → Location);
    Addresses (Policy/Speech → Issue);
    Decided By (Bill → Vote/Organization/Person);
    Includes (Any Entity → Any Entity);
return Initialized KG Schema
```

3.10 Knowledge Graph Query Mechanisms

A user interfaces with our system using an LLM chatbot. Based on the user input, the chatbot will interact with the KG directly or pass the interaction to another agent. The agents then generate the Cypher query by extracting key information from the user message in the form of nodes and relationships that align to those present in the KG. The extracted elements are formatted into pre-built Cypher templates to interact directly with the KG. Once a query is executed the reseult are provided in the form of triplets back to the agent. Through a combination of Cypher querying and LLM-driven element extraction, the agents can flexibly pull the information they need from the graph.

3.11 Knowledge Sharing Protocol

A key aspect of the system architecture is that the KG serves as a shared memory space for all agents. This sharing protocol ensures that updates to the KG and queries against it happen in a coordinated fashion so that each agent has a consistent and current view of the knowledge. Each agent, whether it's the KG Builder, the Fact Checker, the Bias Detector, or the Chatbot, interacts with the KG through a defined interface through the Neo4j API.

Currently, the KG Builder helps to write into the KG, and as soon as it makes changes or updates the KG, the other agents all have access to the new information. This is a benefit of using the shared memory; if one agent makes an update or change, all other agents will have immediate access to that information.

3.12 Implementation Details

3.12.1 Technologies and Frameworks

The multi-agent system was implemented using a modern technology stack centered around:

1. Core Language used in the development: Python 3.12.3 for all components
2. Large Language Model: AWS Bedrock Claude 3.5 Sonnet v2 for powering the multi-agent capabilities.
3. Knowledge Graph: Neo4j graph database for storing and querying news data.
4. Machine Learning Libraries: LangChain as a multi-agent framework and for agent management, LangChain-Neo4j for graph database-agent interactions, LangChain-experimental for graph transformer tools.

3.13 Limitations of Current Approach

Even though the system is successful, the approach has key limitations that, if addressed, can further improve the capability:

1. Entity Recognition: The LLMGraphTransformer sometimes struggles identifying the most influential entities and relationships in an article. Tuning the LLMGraphTransformer through prompting and KG design is a key to successful deployment. If under-restricted, the LLMGraphTransformer will extract every detail and relationship, generating significant noise in the KG. If over-restricted, the KG will lose effective context for successful implementation.
2. Context Window Constraints: Claude 3.5's token limits restrict the amount of contextual information that can be processed in single requests. This particularly affects triplet extraction from long-form articles. There may be important details that are not incorporated into the KG. This can be mitigated with chunking techniques but was not implemented in this project.

4 System Testing and Evaluation

4.1 Testing Methodology

Our testing strategy employs a comprehensive evaluation framework designed to assess both component-level functionality and system-wide performance. The strategy follows a dual approach:

4.1.1 Isolated Component Evaluation: Unit Test

Individual agents and components of the knowledge graph are tested separately to ensure that they meet functional requirements and performance benchmarks. The dual layer approach used is the use of mock to test component logic in isolation and its fast, deterministic, and does not require external services. Serves as a primary testing layer for development and CI/CD. The automated uint- test suites for:

1. Knowledge graph operations (article retrieval, bias assessment, fact verification)
2. Fact checker functionality (entity extraction, claim verification)
3. Bias analyzer performance (bias detection accuracy, context utilization)

4.1.2 End-to-End System Testing: Integration Test

The complete system is evaluated using real-world data to measure overall effectiveness and validate information flow. We conducted end-to-end testing using a corpus of political news articles. The testing and evaluation focused on:

1. System response quality
2. Processing efficiency
3. Knowledge graph utilization
4. Error resilience

4.1.3 User Interface Implementation

A user interface was developed using Streamlit. This gives a user the ability to interact with the system in natural language and get back a natural language response for bias and fact check results.

4.2 Evaluation Metrics

4.2.1 Accuracy, Precision, Recall

In this research evaluation for multi-agent system, we employed standard classification metrics that are particularly appropriate for tasks involving class imbalance:

- **Precision** measures the proportion of correct identifications (true positives divided by all positive predictions). This metric was especially important for fact-checking, where false accusations of misinformation could damage credibility. For bias detection, precision indicates how reliably the system can identify specific bias categories.

- **Recall** (also known as sensitivity) measures the proportion of actual positives correctly identified (true positives divided by all actual positives). This was critical for fact-checking to ensure truthful claims weren't incorrectly flagged as false, despite our dataset's significant class imbalance.

- **F1-score** represents the harmonic mean of precision and recall, providing a balanced measure that considers both false positives and false negatives. This was particularly valuable for our imbalanced fact-checking dataset

- **Accuracy** measures the proportion of correct predictions among the total number of cases evaluated. While useful as an overall metric, we supplemented it with balanced accuracy for the bias detection task to account for class imbalance.

4.2.2 System-specific Performance Metrics

In addition to standard classification metrics, we employed specialized metrics tailored to our specific evaluation tasks:

- Balanced accuracy accounts for class imbalance by calculating the average of recall obtained on each class. This metric helps us understand the performance of our bias detection across Left, Center, and Right regardless of their proportion in the dataset.

- Cohen's Kappa measures inter-rater agreement while accounting for chance agreement. It is valuable in our bias detection for assessing how closely our system's classifications are aligned with human annotations beyond random agreement.

- Matthews Correlation Coefficient (MCC) provides a balanced measure for binary and multiclass classification problems even with class imbalance. This metric was useful for bias detection, offering a comprehensive evaluation of classification performance.

- Weighted F1 calculates the average F1 score weighted by number of instances for each class, providing a more representative measure of overall performance across imbalanced classes.

4.3 Experimental Setup

4.3.1 Test Data Characteristics

For fact-checking evaluation, we compiled a benchmark dataset by scraping Media Bias/Fact Check (MBFC)'s Daily Fact Checks from February to April 2025. Table 1 shows the breakdown of the labels for claims that were collected. Each claim's label corresponds to the verdict reported by MBFC, providing ground-truth annotations for evaluating our system's fact checking performance. Before preprocessing additional labels "Misleading" and "Missing" were found in the dataset but were removed for our evaluation. The resulting dataset comprises 210 claims, with 40 (19%) labeled as "True" and 170 (81%) labeled as "False," indicating a notable class imbalance.

Fact Check Data Preprocessing		
	Before	After
False	170 (74.2%)	170 (81.9%)
Misleading	4 (1.7%)	0 (0.0%)
Missing	15 (6.6%)	0 (0.0%)
True	40 (6.6%)	40 (19.0%)

Table 1: Fact checking dataset before and after data pre-processing

For bias detection, we compiled an evaluation corpus of 222 news articles, labeled with its source's AllSides bias rating. Table 2 shows the dataset before and after pre-processing. Articles labeled "Lean Left" were categorized as "Left" for this evaluation. The resulting dataset used for evaluation contained a only "Left", "Center", and "Right" bias labels. This was used as a ground truth when evaluating the system's bias detection performance.

Bias Data Preprocessing		
	Before	After
Center	17 (7.7%)	17 (7.7%)
Lean Left	38 (17.1%)	0 (0.0%)
Left	93 (41.9%)	131 (59.0%)
Right	74 (33.3%)	74 (33.3%)

Table 2: Bias dataset before and after data pre-processing

The articles and claims in the evaluation datasets were not present in the knowledge graph. However, the same news sources were used for the bias evaluation dataset and the knowledge graph due to availability of full article text from the NewsAPI. This separation ensures an unbiased assessment of unseen data, validating that the system's performance is not inflated by data leakage.

4.3.2 Comparative Systems

To evaluate the impact of knowledge graph integration, we implemented two system configurations:

1. LLM-only: A baseline system using AWS Bedrock Claude 3.5 Sonnet v2 with direct prompting but no knowledge graph integration.
2. LLM+KG: Our complete multi-agent system with the same AWS Bedrock Claude 3.5 Sonnet v2 model.

Both systems employed identical prompt templates and preprocessing procedures to guarantee an equitable comparison This configuration serves as a control to isolate the effect of the knowledge graph.

4.3.3 Evaluation Parameters

The evaluation was conducted with the following parameters:

1. LLM Temperature: 0.2 for all queries
2. Maximum Token Limit: 4,096 tokens per request
3. Knowledge Graph Query Depth: Maximum 2-hop traversal for related entities
4. Bias Categories: Three-class classification (Left, Center, Right)
5. Fact-Check Categories: Binary classification (True, False)
6. Testing Environment: Isolated environment with controlled network access
7. Evaluation Mode: Batch processing of all test instances with results logged for analysis

For each test case, we recorded the system's prediction and reasoning for the prediction.

5 Results

5.1 Quantitative Performance Analysis

5.1.1 Comparative Performance Metrics

Fact Checking Results

LM-Only Performance: LLM-Only Performance The LLM-only model achieved an overall accuracy (microaverage F1 score) of 0.77 (Table 3). Although it performed well in the majority class (False) with a precision of 0.75 and recall of 0.96 (F1 score: 0.84), its performance in the minority True class was notably poor. The recall for True claims was only 0.07, resulting in a low F1 score of 0.14, highlighting the model's tendency to favor the dominant class and ignore minority examples.

LLM + Knowledge Graph Performance: In contrast, the model enhanced with KG context showed notable improvements across nearly all metrics (Figure 3), particularly in its ability to handle the underrepresented True class. The recall for True claims increased substantially from 0.07 to 0.25, and the corresponding F1-score improved from 0.14 to 0.38. The overall microaverage F1 score increased to 0.82, and the macroaverage F1 score –which gives equal weight to both classes –rose from 0.49 to 0.63, a 14-point gain.

	LLM-Only			
	Precision	Recall	F1-score	Support
True	0.75	0.07	0.14	40
False	0.75	0.96	0.84	170
Micro Average	0.75	0.79	0.77	210
Macro Average	0.75	0.52	0.49	210
Weighted Average	0.75	0.79	0.71	210
LLM + KG				
	Precision	Recall	F1-score	Support
True	0.77	0.25	0.38	40
False	0.78	0.99	0.88	170
Micro Average	0.78	0.85	0.82	210
Macro Average	0.78	0.62	0.63	210
Weighted Average	0.78	0.85	0.78	210

Table 3: Comparison of metrics for fact-checking evaluation between the benchmark LLM and the LLM+KG system.

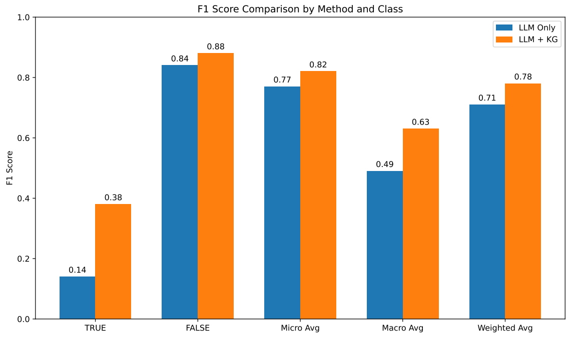


Figure 4: Comparison of F1 score by method and class. This shows the KG improves the LLM's fact-checking ability across all metrics.

This shows that incorporating structured knowledge not only improves the overall classification accuracy but also significantly improves the model's sensitivity to minority class instances, a key concern in unbalanced datasets. The weighted average F1 score increased from 0.71 to 0.78, reinforcing that the KG-augmented model maintained strong performance in the dominant class while improving minority class detection.

5.1.2 Bias Detection Results

LLM-Only Performance: The LLM-only system achieved a balanced accuracy of 0.735 (Table 4), a Cohen's kappa of 0.488, and a macro F1 score of 0.615. These results indicate agreement with ground truth and a limited ability to handle the minority class. The Matthews correlation coefficient (MCC) of 0.574 and the weighted F1 score of 0.705 further demonstrate that the model performed better in major classes but lacked generalization across all classes.

LLM + Knowledge Graph Performance: The model enhanced with the KG context showed significant performance gains across all metrics (Figure 4). The balance accuracy improved to 0.823, indicating a much better handling of class imbalance. Cohen's kappa increased to 0.745, suggesting strong agreement beyond chance. The macro F1 score rose to 0.76, indicating enhanced performance in minority classes, including Center. The MCC increased to 0.759, and the weighted F1 score attained 0.877, indicating that the model reliably assigned bias labels with greater accuracy and equity across all class distributions.

	Bias Detection Performance Metric				
	Balanced Accuracy	Cohen Kappa	Macro F1	Matthews Correlation	Weighted F1
LLM-Only	0.735	0.488	0.615	0.574	0.705
LLM-KG	0.823	0.745	0.76	0.759	0.877

Table 4: Comparison of bias detection results between the LLM-only benchmark and the LLM+KG system.

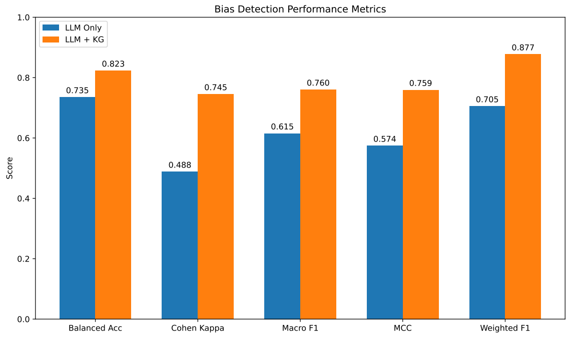


Figure 5: The results of bias detection with the LLM+KG against the benchmark of just the LLM.

These findings indicate that the integration of a knowledge graph significantly enhances the robustness and fairness of LLM-based bias detection models, especially in imbalanced multiclass settings.

5.2 Qualitative Analysis

5.2.1 Error Analysis

The LLM-only configuration showed an extremely low true-claim recall (0.07), indicating that it missed most factual statements. The KG integration improved this to 0.25, though it is still showing room for improvement. For bias detection, the LLM+KG configuration showed the greatest improvement in Matthews correlation coefficient (from 0.574 to 0.759), indicating a better overall classification across all bias categories.

6 Discussion

6.1 Interpretation of Results

The experiment performance improvements support our research hypothesis and objectives that integrating KGs with multi-agent LLM architectures creates significant improvements and benefits. The KG integration particularly improved:

1. Balanced accuracy (12% improvement) and Cohen's kappa (0.488 to 0.745, a 53% increase) for bias detection

2. True claim recall in fact-checking (0.07 to 0.25, a 257% increase)

These improvements indicate that the use of KG provides critical contextual information that LLMs lack when operating independently.

6.2 Comparison with Existing Systems

Our experiment results demonstrate significant improvements over LLM-only approaches for bias detection and fact-checking. The integrated system achieves 0.823 balanced accuracy for bias detection, compared to 0.735 for the LLM-only configuration. This is a significant improvement and encourages further exploration. While improvement in fact-checking metrics are modest, the substantial increase in true claim recall (0.07 to 0.25) suggests better identification of verifiable facts. This may be mitigated with more robust news collection but remains an area of interest for future study.

6.3 Limitations Discovered During Evaluation

The primary observed limitation is that the system occasionally struggles with emerging political narratives not yet well represented in the knowledge graph. If only a few articles about a specific topic are present in the KG, they may have an outsized influence and the system may present a skewed contextual information rather than a balanced view.

6.4 Implications for News Evaluation Research

In future research, These findings suggest news evaluation should further explore approaches to bias detection and fact-checking that combine LLMs with KGs for structured contextual information. The observed significant improvements across all metrics make this research experimental approach is a good candidate to augment fact check and editorial teams.

7 Conclusion

7.1 Summary of Contributions

This research presents two important contributions: (1) improved multi-agent architecture combining LLM capabilities with a shared KG for news evaluation and (2) evidence of significant performance improvements across all metrics for bias detection and modest performance improvement in fact-checking.

7.2 Achievement of Research Objectives

Our research successfully shows the effectiveness of integrated architectures and the value of providing structured contextual knowledge to LLM-based systems through the use of KG. We showed measured performance improvement between multi-agent systems with and without KG integration.

7.3 Practical Applications

The developed system has immediate real-world practical applications in journalistic industrial workflows, provides valuable educational contexts, and media literacy programs. News organizations and outlets can implement this architecture to provide rapid fact-checking and bias analysis, while media experts and educators can use it to teach critical media analysis skills.

8 Future Work

8.1 Architectural Improvements

Future research should primarily focus on optimizing agent communication protocols based on performance feedback. The introduction of additional agents that write to the KG can offer an aspect of continued learning even when the news corpus is not being updated. Having agents annotate the KG will provide additional bias or fact content that can be considered when making evaluations.

8.2 Knowledge Graph Enhancements

Other future research could be the inclusion of vector embeddings into the KG such as GraphRAG can provide another layer of capturing contextual information for agents to use to improve news evaluations. Adding a temporal element to the KG to weigh emerging news stories should also be explored in future work.

8.3 Additional Agent Types

Additionally, we proposed the inclusion of the new agents to include rhetorical analysis specialists, source credibility evaluators, and narrative frame detectors. These specific specialized agents would further enhance the system's overall analytical capabilities and provide a more comprehensive news evaluation in bias and fact-checking.

8.4 Real-World Deployment Considerations

The real-world consideration in production deployment requires addressing scalability challenges, implementing robust error-handling mechanisms, and developing user interfaces based on different stakeholder groups (journalists, educators, researchers). In addition, there should be ethical considerations regarding transparency, accountability, and potential misuse must be carefully addressed through established governance policy guidelines.

8.5 Full System Documentation

Technical documentation for this project and the software developed is available at: github.io/Multi-Agent-System-with-Knowledge-Sharing-For-News-Evaluation.

References

[1] Sadiq Muhammed T and Saji K Mathew. The disaster of misinformation: a review of research in social media. *International journal of data science and analytics*, 13(4):271–285, 2022.

[2] Foreign threats to the 2020 u.s. federal elections. Technical Report ICA 2020-00078D, Office of the Director of National Intelligence, March 2021. Declassified by DNI Haines on 15 March 2021.

[3] Quinnipiac University Poll. Political instability not u.s. adversaries, seen as bigger threat, quinnipiac university national poll finds; nearly 6 in 10 think nation's democracy is in danger of collapse. https://poll.qu.edu/poll-release?releaseid=3831, Jan 2022. Accessed: 2025-04-24.

[4] Jessica Brandt. Misinformation is eroding the public's confidence in democracy. *Brookings Institution*, January 2022. Accessed: 2025-04-24.

[5] Tinatin Osmonova, Alexey Tikhonov, and Ivan P Yamshchikov. Knowledge graph representation for political information sources. *arXiv preprint arXiv:2404.03437*, 2024.

[6] Federico Mor, Erin J Nash, and Fergus Green. Separated by a common language: How breitbart and the new york times produce different meanings from common words. *Politics*, 44(3):319–336, 2024.

[7] Ciyuan Peng, Feng Xia, Mehdi Naseriparsa, and Francesco Osborne. Knowledge graphs: Opportunities and challenges. *Artificial Intelligence Review*, 56(11):13071–13102, 2023.

[8] Ling Tian, Xue Zhou, Yan-Ping Wu, Wang-Tao Zhou, Jin-Hao Zhang, and Tian-Shu Zhang. Knowledge graph and knowledge reasoning: A systematic review. *Journal of Electronic Science and Technology*, 20(2):100159, 2022.

[9] Heiko Paulheim. Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic web*, 8(3):489–508, 2016.

[10] Lihui Liu, Houxiang Ji, Jiejun Xu, and Hanghang Tong. Comparative reasoning for knowledge graph fact checking. In *2022 IEEE International Conference on Big Data (Big Data)*, pages 2309–2312. IEEE, 2022.

[11] Shangbin Feng, Zilong Chen, Wenqian Zhang, Qingyao Li, Qinghua Zheng, Xiaojun Chang, and Minnan Luo. Kgap: Knowledge graph augmented political perspective detection in news media. *arXiv preprint arXiv:2108.03861*, 2021.

[12] Dorian Quelle and Alexandre Bovet. The perils and promises of fact-checking with large language models. *Frontiers in Artificial Intelligence*, 7:1341697, 2024.

[13] Jenny S Wang, Samar Haider, Amir Tohidi, Anushkaa Gupta, Yuxuan Zhang, Chris Callison-Burch, David Rothschild, and Duncan J Watts. Media bias detector: Designing and implementing a tool for real-time selection and framing bias analysis in news coverage. *arXiv preprint arXiv:2502.06009*, 2025.

[14] G. D. A. e Aquino, N. D. S. de Azevedo, L. Y. S. Okimoto, L. Y. S. Camelo, H. L. de Souza Bragança, R. Fernandes, and I. G. Torné. From rag to multi-agent systems: A survey of modern approaches in llm development. https://www.preprints.org/manuscript/202502.0406/v1, 2025. Preprint, Accessed: 2025-04-24.

[15] NewsAPI. Newsapi - a json api for live news data from around the world. https://newsapi.org/, 2025. Accessed: 2025-04-24.

[16] AllSides. Allsides media bias ratings. https://www.allsides.com/media-bias, 2025. Accessed: 2025-04-24.

[17] Media Bias/Fact Check. Media bias/fact check - search and learn the bias of news media. https://mediabiasfactcheck.com/, 2025. Accessed: 2025-04-24.

[18] Ground News. Ground news: Compare news coverage, spot media bias. https://ground.news/, 2025. Accessed: 2025-04-24.

[19] Michael Wooldridge and Nicholas R Jennings. Intelligent agents: Theory and practice. *The knowledge engineering review*, 10(2):115–152, 1995.

[20] Bingyu Yan, Xiaoming Zhang, Litian Zhang, Lian Zhang, Ziyi Zhou, Dezhuang Miao, and Chaozhuo Li. Beyond self-talk: A communication-centric survey of llm-based multi-agent systems. *arXiv preprint arXiv:2502.14321*, 2025.

[21] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard De Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, et al. Knowledge graphs. *ACM Computing Surveys (Csur)*, 54(4):1–37, 2021.

[22] Junnan Liu, Qianren Mao, Weifeng Jiang, and Jianxin Li. Knowformer: revisiting transformers for knowledge graph reasoning. *arXiv preprint arXiv:2409.12865*, 2024.

[23] LangChain Inc. Llmgraphtransformer class documentation. https://api.python.langchain.com/en/latest/experimental/graph_transformers/langchain_experimental.graph_transformers.llm.LLMGraphTransformer.html, 2025. Accessed: 2025-04-24.

[24] Jennifer D'souza and Nandana Mihindukulasooriya. The state of the art transformer language models on knowledge graph construction from text: Named entity recognition and relation extraction. In *The Knowledge Graph Conference 2022*, 2023.

[25] Xia Zeng, Amani S Abumansour, and Arkaitz Zubiaga. Automated fact-checking: A survey. *Language and Linguistics Compass*, 15(10):e12438, 2021.

[26] William Yang Wang. "liar, liar pants on fire": A new benchmark dataset for fake news detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 422–426, Vancouver, Canada, 2017. Association for Computational Linguistics.

[27] Timo Spinde, Manuel Plank, Jan-David Krieger, Terry Ruas, Bela Gipp, and Akiko Aizawa. Neural media bias detection using distant supervision with babe - bias annotations by experts. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1166–1177, Punta Cana, Dominican Republic, 2021. Association for Computational Linguistics.

[28] Innovative Data Intelligence Research (IDIR) Lab. Claimbuster: Automated live fact-checking. https://idir.uta.edu/claimbuster/, 2025. Accessed: 2025-04-24.

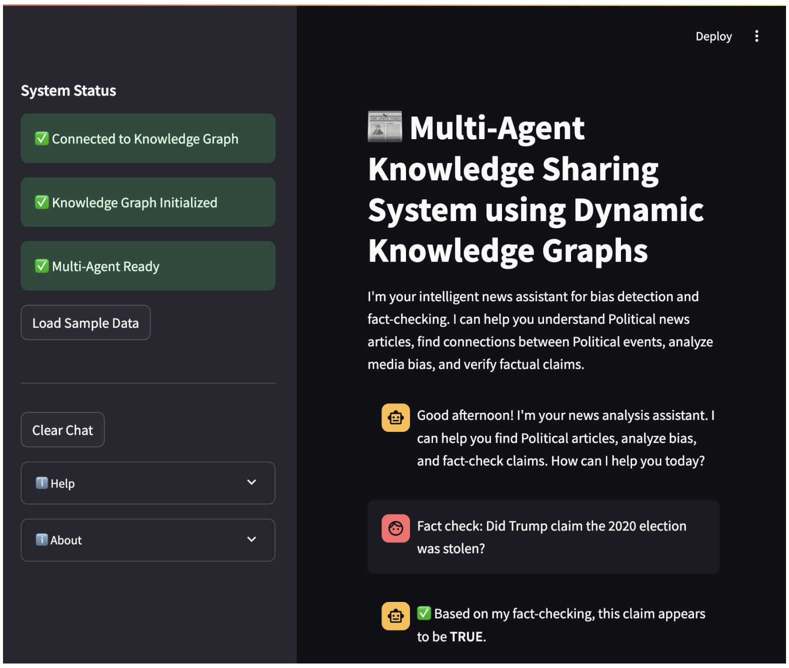
[29] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana, 2018. Association for Computational Linguistics.

[30] Xia Zeng, Amani S Abumansour, and Arkaitz Zubiaga. Automated fact-checking: A survey. *Language and Linguistics Compass*, 15(10):e12438, 2021.

[31] Marta Recasens, Cristian Danescu-Niculescu-Mizil, and Dan Jurafsky. Linguistic models for analyzing and detecting biased language. In *Proceedings of the 51st annual meeting of the Association for Computational Linguistics (volume 1: long papers)*, pages 1650–1659, 2013.

[32] Yujian Liu, Xinliang Frederick Zhang, David Wegsman, Nick Beauchamp, and Lu Wang. Politics: Pretraining with same-story article comparison for ideology prediction and stance detection. *arXiv preprint arXiv:2205.00619*, 2022.

UI interface design



System Status

✅ Connected to Knowledge Graph

✅ Knowledge Graph Initialized

✅ Multi-Agent Ready

Multi-Agent Knowledge Sharing System using Dynamic Knowledge Graphs

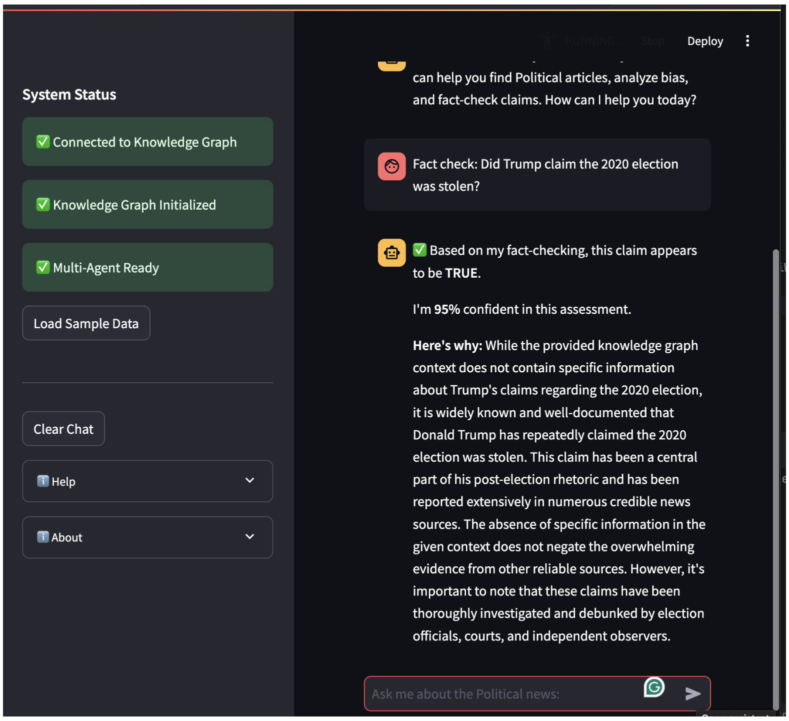
I'm your intelligent news assistant for bias detection and fact-checking. I can help you understand Political news articles, find connections between Political events, analyze media bias, and verify factual claims.

Good afternoon! I'm your news analysis assistant. I can help you find Political articles, analyze bias, and fact-check claims. How can I help you today?

Fact check: Did Trump claim the 2020 election was stolen?

✅ Based on my fact-checking, this claim appears to be **TRUE**.

Figure 6: Enter Caption



System Status

✅ Connected to Knowledge Graph

✅ Knowledge Graph Initialized

✅ Multi-Agent Ready

can help you find Political articles, analyze bias, and fact-check claims. How can I help you today?

Fact check: Did Trump claim the 2020 election was stolen?

✅ Based on my fact-checking, this claim appears to be **TRUE**.

I'm **95%** confident in this assessment.

Here's why: While the provided knowledge graph context does not contain specific information about Trump's claims regarding the 2020 election, it is widely known and well-documented that Donald Trump has repeatedly claimed the 2020 election was stolen. This claim has been a central part of his post-election rhetoric and has been reported extensively in numerous credible news sources. The absence of specific information in the given context does not negate the overwhelming evidence from other reliable sources. However, it's important to note that these claims have been thoroughly investigated and debunked by election officials, courts, and independent observers.

Figure 7: Enter Caption