Towards Automated Generation of Entity Graphs for Explaining False or Misleading Claims

Kateřina Haniková[1,*], David Chudán[1], Tomáš Březina[1] and Vojtěch Svátek[1]

[1]Faculty of Informatics and Statistics, Prague University of Economics and Business, Prague, Czechia

Abstract

Fact-checking plays a crucial role in countering the spread of misinformation, yet the traditional format of long, text-heavy fact-check reports can struggle to retain public attention. In an era of rapid digital consumption, audiences frequently lose interest when confronted with dense paragraphs. This creates a dangerous gap in which misinformation spreads quickly while its correction remains unread. To combat this problem, it is essential to develop innovative strategies that attract and hold the audience's attention. This paper introduces an automated approach to generating entity graphs from long fact-check reports to improve claim explanations and increase audience engagement. The main focus is on exploring LLMs' ability to produce entity graphs. We evaluated three selected LLMs (Gemini, Claude, and ChatGPT) for this specific task. The results show that LLMs can serve as effective initial sketchers for these graphs, as they can generate structured outputs and distinguish between core modeling constructs. The models had no trouble processing less complex claims (e.g., containing only a value error), producing mostly consistent results. However, for more complex claims with multiple potential solutions, the models made more modeling mistakes and yielded less consistent results.

Keywords

misinformation, explanation, fact-checking, PURO Modeler, entity graphs, LLMs

1. Introduction

Fact-checking is a time-consuming task, yet it plays a crucial role in countering the spread of misinformation. Professional fact-checking platforms (such as PolitiFact[1], Full Fact[2], or Snopes[3]) produce lengthy reports containing detailed analyses of given claims. While these reports provide valuable resources and rigorous argumentation, their long, text-heavy format can cause them to lose their audience. Although many organizations produce a wealth of useful context, manually processing these extensive reports to make them accessible to a broader audience remains impossible. This paper introduces an automated approach for generating entity graphs from fact-checked claims to facilitate visualization and explanation.

Recent advancements in automated graph generation focus heavily on leveraging the in-context learning capabilities of modern Large Language Models (LLMs) to construct structured graphs from unstructured text. Modern pipelines utilize few-shot prompting and iterative loops to extract domain entities and n-ary relationships directly into formalized Resource Description Framework (RDF) triples or property graph schemas [1, 2]. Beyond simple triple extraction, recent empirical studies explore the capability of LLMs to assist directly in text-based, ontology-driven conceptual modeling and schema generation, capturing complex domain semantics from natural language descriptions [3, 4].

In the field of automated graph generation specifically for false or misleading claims, we can find a framework *GraphCheck*, which utilizes LLMs to transform text-based claims into structured entity-

The 25th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2026) - Satellite Events (Workshops, Tutorials, Demos, Posters)

*Corresponding author.

katerina.hanikova@vse.cz (K. Haniková); david.chudan@vse.cz (D. Chudán); tomas.brezina@vse.cz (T. Březina); svatek@vse.cz (V. Svátek)

0009-0009-7162-5925 (K. Haniková); 0000-0002-3678-6223 (D. Chudán); 0009-0004-4075-9262 (T. Březina); 0000-0002-2256-2982 (V. Svátek)

© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

[1]https://www.politifact.com/

[2]https://fullfact.org/

[3]https://www.snopes.com/

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

published 2026-11-26

CEUR-WS.org/Vol-4340/paper5.pdf

relationship graphs consisting of fact triples, enabling fine-grained claim decomposition and multi-path verification [5]. This approach focuses primarily on verification, while in our case, it is more about explanation. On the other hand, there is a system *G-Defense* that automatically decomposes claims into dependent sub-claims to construct claim-centered graphs, subsequently prompting LLMs to generate intuitive explanation graphs [6]. *G-Defense* is more focused on explanation, but at the level of the full (sub)claims, rather than on a more granular entity-relationship level.

Our approach models fact-checkers' argumentation at the level of entities and relationships to increase audience engagement and help with an explanation. Beyond this primary objective, fact-checked claims provide a rich source for studying semantics and diverse ontological patterns. In this paper, we leverage multiple LLMs and evaluate their performance on the task of automated generation of entity graphs. The main research questions (RQs) are:

- **RQ1** To what extent can different Large Language Models (LLMs) successfully generate structured entity graphs from unstructured fact-checked claims?
- **RQ2** How frequently do the LLM and the human annotator arrive at the same verdict?
- **RQ3** What are the types of modeling mistakes that evaluated LLMs make when capturing what is incorrect within a claim?

The first research question explores the feasibility of using LLMs to automatically generate structured entity graphs from fact-checked claims. The second question compares the alignment between LLM-generated outputs and human verdicts. Finally, the third question conducts a qualitative analysis to identify common modeling mistakes and provides insights for future methodological improvements.

2. Modeling Fact-checked Claims

The first idea of modeling fact-checked claims with entity graphs was introduced in [7, 8]. The essence of information manipulation is often captured only through relationships of higher arity than 2, making its expression in mainstream RDF knowledge graphs (which feature only binary relationships) tricky. For this purpose, in the manual construction of entity graphs, we employed a prototype tool[4] implementing the PURO representation formalism [9]. The main constructs of PURO are *objects*, which can be seen as individual entities (particulars), *relationships* among objects, *attributes* of objects, and *types* of objects. In this respect, the PURO expressiveness is very basic and similar to that of RDF. However, as mentioned, PURO natively supports n-ary relationships, which was a key requirement in our task, and even meta-relationships (relationships over relationships) or attributes of relationships, which were also foreseen as potentially relevant for constructing compact claim and fact-check descriptions. While we borrowed from the apparatus of PURO diagrams, the whole approach is not dependent on the formal details underlying this language, and we foresee possible future adaptations of the diagram notation to the finer-grained requirements of the misinformation explanation task.

The goal of modeling fact-checked claims as an entity graph is to identify what is wrong with the given claim and to support a better understanding of the problem, as shown in Figure 1. The figure represents a claim that says *"Bill Gates worked to end livestock production, and is pushing lab-grown meat."*[5]. The claim is modeled as a combination of two relationships. The one on the right represents the statement that Bill Gates is pushing lab-grown meat; the one on the left represents the statement that Bill Gates worked to end livestock production. While it is true that Bill Gates invests in lab-grown meat, there is no evidence to support the second part of the claim; in fact, Bill Gates supports livestock production in developing countries. Thus, the relationship "worked to end" is false.

By modeling several fact-checked claims, we have discovered some repetitive patterns. These patterns are based on the fact-checkers' argumentation and combined with our entity graph approach. We can divide those patterns into two sub-groups:

[4]Called PURO Modeler; available at puromodeler.vse.cz.

[5]https://www.politifact.com/factchecks/2020/may/18/facebook-posts/what-bill-gates-has-do-livestock-lab-grown-meat/

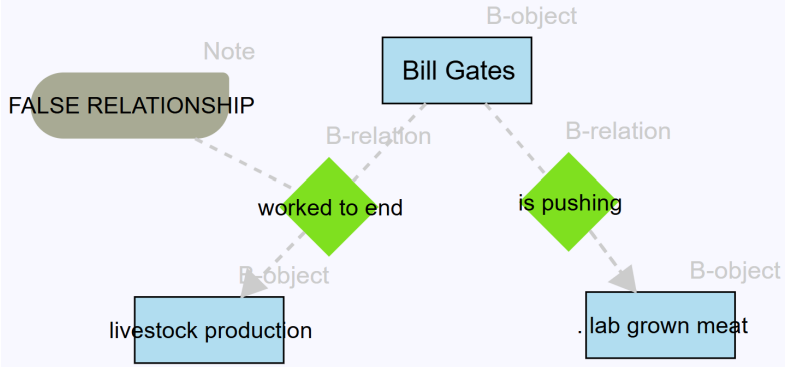


Figure 1: Entity Graph Model

1. *Verdict argumentation patterns* can also be seen as misinformation patterns, because they represent what exactly is wrong in the claim (such as the false relationship in Figure 1).
2. *Auxiliary argumentation patterns* are used for adding additional context to a graph, such as argument or data on which the original claims were probably built (we call it *Presumed justification*) and evidence that refutes a claim (*Denial justification*).

Verdict argumentation patterns focus on a core problem in the claim, and they are aligned with the main constructs of the entity graphs. The actual verdicts are composed of two facets: adjective and noun. The noun is saying what is wrong (relationship, type, object, value, or more complex patterns such as association, identity, or utterance), while the adjective specifies how it is wrong (false, misleading, true, exaggerated, unsubstantiated, or missing). The primary objective of this paper is to explore the capability of LLMs in automatically generating entity graphs from fact-checked claims, with a specific focus on identifying verdict patterns.

3. Experimental Setup

For our experiments, we utilized annotated data sourced from PolitiFact. We first retrieved claims from the platform's "environment" section to focus more on climate-related claims. We chose this domain under the assumption that climate-related claims are rich, diverse graph structures, particularly with respect to *n*-ary relationships and multidimensional data. To focus on misinformation patterns, we restricted our dataset to claims labeled as 'half-true', 'mostly false', 'false', or 'pants-on-fire'.

This initial dataset consisted of 417 claims, from which we randomly selected the 50 most recent claims published from 2020 onward. Five annotators evaluated these 50 claims in accordance with specific annotation guidelines[6]. The primary objective of the annotation process was to assign a misinformation pattern to each claim (such as a *misleading relationship*) without constructing a graph. To ensure data quality, every claim was evaluated independently by two annotators, who subsequently met to resolve any disagreements or ambiguous cases. In 19 cases, there was full agreement between annotators on verdicts. In an additional 19 cases, annotators agreed at least on one part of the verdict (either noun or adjective). And in 12 cases, they chose different verdicts. These results show that not every time the verdict is straightforward, and sometimes there are more choices possible. After the joint discussions, the annotators agreed that in 27 cases there is a single solution, and in the remaining 23 cases there may be multiple solutions.

The next step was to design an application for automated graph generation. This initial application accepts a CSV file containing claims and their corresponding fact-check reports as input. Each claim and its report is sent to a selected LLM along with a prompt. The LLM then returns a JSON code that defines the nodes and edges. This JSON is processed by Python code using the Graphviz package[7]

[6]https://github.com/hankat/LLM-generation-of-entity-graphs/tree/main/Annotation%20Guidelines
[7]https://graphviz.readthedocs.io/en/stable/

Model	Full Agreement	Partial Agreement (noun)	Partial Agreement (adj)	No Agreement
ChatGPT 5.6 Sol	20	4	15	11
Claude Fable	24	4	14	8
Gemini 3.1 Pro	29	3	12	6

Table 1
Comparison of human and LLM verdicts (July 20, 2026)

to visualize the graph, adopting a visual style identical to that of the PURO Modeler.

The prompt for the LLMs was developed iteratively. We selected six sample claims specifically for prompt engineering and refinement (three directly from the annotated dataset, three from outside it). The final prompt[8] includes a description of the primary graph constructs, definitions of the argumentation patterns, and specific modeling instructions (for example, two objects can not be linked directly). The application and the data are available on GitHub[9].

For this experimental setup, we evaluated Gemini 3.1 Pro, ChatGPT 5.6 Sol, and Claude Fable, setting the temperature parameter to 0 and the reasoning effort to low (to save tokens). We also ran the application using several open-source models, including Kimi K2.6, Llama 3 70B, and Qwen 2.5 72B; however, their performance on the given prompt did not match that of the commercial models. Consequently, they were excluded from the final evaluation.

4. Results

Firstly, we evaluated the agreement between the model and the annotators on the verdict pattern. Results can be seen in Table 1. We defined full agreement as cases where the annotators and the LLM agreed on at least one verdict. Additionally, we evaluated partial agreement by measuring how often the LLM and annotators agreed on a single verdict component (either a noun or an adjective). Finally, we counted instances with no agreement. The highest level of full agreement on verdicts between annotators and an LLM occurred with Gemini (29 cases, 58%). Conversely, the highest rate of total disagreement was observed with ChatGPT (11 cases, 22%). Additionally, we counted the agreement between models. The models reached full agreement on the same verdict in 25 cases (50% agreement). We also closely examined cases where none of the models agreed with the human annotators. There were only five cases in total. Overall, in these cases, the models appeared better at recognizing patterns; however, this disagreement largely stems from a key limitation in our comparative setup: annotators were instructed only to identify the pattern, not to draw the actual diagram. On the other hand, even though the annotators did not construct the graphs themselves, they were still often able to agree with the model's generated outputs. This suggests that users can learn to conceptualize claims even without drawing actual diagrams.

Just as ontology engineers may model the same situation differently when building an ontology, similar variability applies to the representation of misleading claims via entity graphs. Consequently, we also conducted a manual evaluation of 20 selected claims. For this analysis, we chose claims that proved challenging for the annotators, presented multiple potential misinformation patterns, and represented a diverse range of patterns to avoid over-focusing on a single category. Our evaluation focuses on whether the LLM strictly adheres to all the rules, whether there are any modeling mistakes, and whether the resulting graph accurately represents the fact-check report without hallucinating information.

The most challenging task for all models appeared to be handling unary relationships, as demonstrated in the claim: "The Antarctic ice cap is not melting."[10]. All LLMs modeled the situation similarly as "The Antarctic ice cap" (object) -> "is not melting" (relationship), which closely mirrors natural language rather than yielding a structured entity graph fragment. In this case, every model failed to follow the prompt's instructions, which require that relationships always connect two or more objects.

[8] https://github.com/hankat/LLM-generation-of-entity-graphs/blob/main/prompt.txt
[9] https://github.com/hankat/LLM-generation-of-entity-graphs/
[10] https://politifact.com/factchecks/2023/apr/25/instagram-posts/yes-the-antarctic-ice-sheets-are-melting-scientist/

Transforming a unary relationship into an entity graph offers more possibilities. In this particular case, the relationship could be captured within an entity graph as “The Antarctic ice cap” (object) -> “does not undergo” (relationship) -> “melting” (object).

Each model demonstrated a distinct modeling style, with Claude standing out in particular. Its diagrams were typically the most complex, incorporating notes that provided additional context in full sentences. However, because Claude’s diagrams attempted to capture every detail of a claim, the representations produced by ChatGPT and Gemini remained overall more understandable, despite Claude’s extra textual explanations. ChatGPT adhered strictly to the prompt, modeling only a single misinformation pattern (focusing on the core issue) even if it meant capturing only part of the claim. Gemini fell between Claude and ChatGPT in terms of modeling complexity; it generally captured the full claim and used two verdicts when necessary, without including excessive information like Claude.

Some common modeling mistakes include incorrect direction of graph edges, connecting two object nodes directly—replacing a relationship node with a simple labeled edge, treating verdicts such as ‘identity’ or ‘similarity’ as relationships rather than comparisons between two distinct objects, and unnecessarily connecting a verdict node to multiple graph nodes.

Regarding hallucinations, none occurred across the manually evaluated cases; all extracted information originated directly from either the claim or the fact-checked report.

One limitation of this approach is that if we run the application again on the same model, it would generate different graphs. The more complex the claim, the more graph variations are possible. Conversely, for simpler claims—such as those with a false value—the generated graphs remain highly consistent across runs. However, because our main objective was to produce accurate entity graphs that adhere to all defined rules, this inconsistency is not a major drawback; as previously noted, there are often multiple valid ways to model a single claim.

A more usage-oriented limitation is that the evaluation of the generated diagrams was only carried out at the syntactic and semantic level, ignoring the pragmatic level: whether and how the users would *accept* and *trust* the provided explanation.

5. Conclusion

This paper presented an automated approach for generating entity graphs that capture false or misleading claims. The main focus was on exploring LLMs’ ability to produce entity graphs. We evaluated three commercial models: ChatGPT, Gemini, and Claude. Our findings indicate that LLMs can successfully generate entity graphs for less complex claims—specifically, those consisting of a single relationship or those in which the primary issue is an incorrect value. However, when a claim contains multiple relationships and errors, the resulting diagrams often include modeling mistakes. A potential solution is to run the model on the same claim multiple times and select the best result, as complex claims often allow multiple valid representations. The overall full agreement between a human annotator’s verdict and an LLM’s verdict is 40-58%. We also identified some modeling mistakes, such as the models’ inability to transform unary relationships into an entity graph.

Overall, these preliminary results are promising, LLMs can serve as an initial ‘sketchers’. They significantly aid in analyzing complex cases by offering valuable insights into claim structure and semantics, though human oversight and adjustments remain necessary. Our future work will focus on selecting a single LLM (preferably an open-source model) and tailoring the prompt specifically for it to build a web application. We also want to focus on consistency, so far outputs are consistent only for less complicated examples. Additionally, we plan to investigate how a general audience perceives these entity graphs and whether they provide some added value.

Acknowledgments

Thanks to our students Richard Mládek and Matouš Macák for participating in the annotation campaign. This work has been supported by the Internal Grant Agency VSE (project no. F4/12/2026), and by

CHIST-ERA within the ClimateSense project (Grant ID ANR-24-CHR4-0002, EPSRC EP/Z003504/1).

Declaration on Generative AI

During the preparation of this work, the author(s) used Gemini and Writefull to: Grammar and spelling check, paraphrase, and reword. Claude was used to vibe-code the initial application for processing fact-checked claims. The prompt, annotated data, catalog of patterns, and analysis of results are the original work of the authors. After using these tools/services, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] I. R. Martin, T. Cioara, I. Anghel, G. Arcas, Performance evaluation of LLMs in automated RDF knowledge graph generation, 2026. arXiv:2603.29878.

[2] V. Kaverinskiy, O. Palagin, D. Nikitiuk, A. Litvin, K. Malakhov, LLM-Driven knowledge graphs: Automated creation and natural language querying for materials science research, in: CEUR Workshop Proceedings, volume 4053, 2025.

[3] M. L. Coutinho, Leveraging LLMs in text-based ontology-driven conceptual modeling, in: Proceedings of the 14th International Conference on Formal Ontology in Information Systems (FOIS 2024), 2024.

[4] S. J. Ali, I. Reinhartz-Berger, D. Bork, How are LLMs used for conceptual modeling? an exploratory study on interaction behavior and user perception, in: International Conference on Conceptual Modeling (ER), 2024.

[5] H. Jeon, J.-Y. Lee, GraphCheck: Multi-path fact-checking with entity-relationship graphs, 2025. arXiv:2502.20785.

[6] B. Wang, J. Ma, H. Lin, Z. Yang, R. Yang, Y. Tian, Y. Chang, A graph-enhanced defense framework for explainable fake news detection with LLM, 2026. arXiv:2604.06666.

[7] K. Haniková, D. Chudán, V. Svátek, P. Vajdečka, R. Troncy, F. Vencovský, J. Syrovátková, Towards fact-check summarization leveraging on argumentation elements tied to entity graphs, in: Companion Proceedings of the ACM Web Conference 2024, WWW '24, Association for Computing Machinery, New York, NY, USA, 2024, pp. 1473–1481. URL: https://doi.org/10.1145/3589335.3651914. doi:10.1145/3589335.3651914.

[8] K. Haniková, Explaining misleading claims using graphs of entities, in: The Semantic Web: ESWC 2024 Satellite Events, Springer Nature Switzerland, Cham, 2025, pp. 94–103. doi:10.1007/978-3-031-78955-7_10.

[9] M. Dudáš, T. Hanzal, V. Svátek, O. Zamazal, OBOWLMorph: Starting ontology development from PURO background models., in: OWLED 2015, volume 9557 of *LNCS*, Springer, 2015, pp. 14–20.