From Text to Structure: Extracting and Validating Complex System Representations Using Large Language Models

Carlo Lipizzi

School of Engineering and Science - Dpt. of Systems Engineering, Stevens Institute of Technology, Hoboken, NJ, USA

Abstract: Managing complexity in systems engineering requires accurate representations of the system's structure and dependencies. This work introduces a semi-automated/human-in-the-loop method to extract and validate structural representations of complex systems from textual descriptions using Large Language Models (LLMs). The proposed pipeline consists of: (1) collecting relevant system documentation, (2) extracting subject–predicate–object triplets using an LLM, (3) building a graph where nodes are the subjects and the objects - representing system entities - and edges are the predicates - representing relationships, and (4) conducting Subject Matter Expert (SME) validation of the resulting graph. This verified graph serves as a formal structural model of the system. The graph could be used as such as well as to assess the consistency of future LLM-generated responses.

Keywords: Dependency and Structure Modelling, Knowledge Graph, System Representation, Large Language Models, Triplet Extraction, Vector Similarity, AI Validation

1. Introduction

Representing a system with a model is critical for effectively managing systems in general and complex systems in particular. Models provide an abstracted yet structured view of its components, interactions and dependencies. These representations serve as essential tools for understanding complexity, predicting behavior and informing decision-making in engineering and operational contexts. However, the usefulness of a model hinges on its trustworthiness, that is the degree to which stakeholders can rely on it to accurately and consistently reflect the real-world system it represents. A trustworthy model must not only capture relevant elements of the system with fidelity but also be constructed and validated in a transparent and repeatable way. In this sense, trustworthiness becomes a key attribute of the representation itself, influencing its adoption and utility in critical domains where decisions carry high stakes, such as aerospace, healthcare and infrastructure. Ensuring that a model is trustworthy involves rigorous validation, ideally involving both automated methods and expert review, to confirm that the structure and semantics of the model align with the actual system's behavior and constraints.

As supported by vast literature (for example (Holsapple, 2008)), we base our decision on knowledge. This knowledge can be from direct experience or from sources we rely on. Sources can be humans - such as subject matter experts - or from tools we "trust".

A range of studies have explored the use of knowledge tools in decision-making. (Boose et al., 1992) and (Brown, 1989) both emphasize the importance of comprehensive decision models, with Boose specifically advocating for the use of knowledge acquisition techniques. (H. Kopackova et al., 2007) and (Rundall et al., 2007) focus on preprocessing tools and evidence-informed decision-making, respectively, with Rundall proposing the Informed Decisions Toolbox. (Ytsen Deelstra et al., 2003) and (Yost et al., 2014) both stress the need for a strong connection between knowledge and decision-making, with Yost providing a practical example of tools used in public health departments. (Yavuz et al., 2005) takes a more pragmatic approach, using a knowledge-based system to develop a marketing decision model. Collectively, these studies highlight the potential of knowledge tools in enhancing decision-making processes.

The method presented in this paper is a novel approach to create a trustworthy model with a semi-automated process using a combination of Large Language Models, Knowledge Graphs and RDF triplets (subject, predicate, object), with humans in the loop.

The prototype that has been developed shows encouraging results.

2. Literature Review

A range of studies have explored the measurement of trust in large language models (LLMs). (Koehl & Vangsness, 2023) highlight the potential of LLMs in analyzing qualitative responses. This is an interesting perspective, but it does not address the problem of getting a quantitative evaluation of bias in LLMs. (Yang Liu et al., 2023) focus on the importance of alignment with human intentions. (Yue Huang et al., 2023) have an interesting approach to evaluating LLMs in three crucial areas: toxicity, bias, and value alignment. Their approach ("TrustGPT") is focused on ethical and moral compliance of LLMs, which is a relevant one, but it leaves out the cases where "trust" overlaps with "accuracy" for a given domain or subject.

(L. Sun et al., 2024) is a massive study ("TrustLLM") conducted by 23 researchers defining trustworthiness for an LLM as a combination of truthfulness, safety, fairness, robustness, privacy, machine ethics, transparency and accountability. They measure the answers provided by a given LLM against "gold answers" from given datasets. While this approach has evident merits, it does not address the problem of measuring the trustworthiness related to the specific domain the user is interested in.

Several other studies are primarily pointing to the problem without providing guidance for measuring the actual trustworthiness. For example, (P. Bhandari & H. M. Brennan, 2023) find that LLMs struggle to generate high-quality children's stories, and (Jacob Menick et al., 2022) emphasize the need for models to support answers with verified quotes. (Rick Rejeleene et al., 2024) have a more comprehensive approach, recognizing the importance of trust ("Trust plays a central role in economic transactions, for the majority of professions in businesses") and drilling down on the reasons why LLMs have trust issues. They base the evaluation of the quality of LLMs on three dimensions: accuracy, consistency and relevance. Their theoretical approach analyzes the logical/mathematical limitations of the algorithms and processes generating the LLMs. No actual measurement of the quality is provided.

Similarly to the previously cited authors, (Rachith Aiyappa et al., 2023) underscore the complexity of measuring trust in LLMs, analyzing the complexity of the process of creating an LLM and manually testing the accuracy of some LLMs to highlight the need for ongoing research in this area. They also cast a shadow on the existing methods to evaluate the performance of those models, often based on testing datasets that could be part of the datasets used to train the models, going against basic rules of data science (separation of training and testing) and common sense ("You can't judge a contest in which you are a participant").

Another element of complexity in the evaluation of trustworthiness is its subjectivity. While the fundamental concept of trust as a belief in the reliability and integrity of others (or systems) is a universal aspect of human life, the expression and significance of trust can vary widely across different cultures and contexts. For instance, collectivist cultures may emphasize consensus-based validation, while individualist cultures might prioritize technical accuracy. These factors influence how SMEs interpret and validate models generated by LLMs

(Rosanas, 2004) highlights the relevance of the "trustor" and the "trustee". Referring to the process of delivering trust, "The situation involves two decision makers, A (the trustor) and B (the trustee)". This echoes a previous position expressed by (Gambetta, 2000), who defines trust as "the subjective probability with which an agent expects that another agent or group of agents will perform a particular action on which its welfare depends".

3. Objective

Leveraging on LLMs could improve the efficiency and effectiveness of generating models representing the system. Building such systems requires more than post hoc evaluation. It demands processes that ensure alignment with accurate, domain-relevant knowledge. While human evaluators have traditionally played a role in interpreting LLM outputs, their involvement often introduces variability and limits scalability. Due to the difficulties of accounting factors such as common sense and logic, human evaluators are often involved in analyzing computer-generated output (Goodrich et al., n.d.). This is raising issues of consistency (Clark et al., n.d.) and scalability.

This paper proposes a structured, semi-automated approach to constructing trustworthy systems by embedding domain knowledge into formal representations through LLMs and validating them with expert oversight.

This paper introduces a novel process to embed trustworthiness into the model by converting unstructured textual documentation into structured, Subject Matter Expert (SME)-validated graphs using subject–predicate–object triplets extracted from text representing a ground truth for the system's requirements.

4. Methodology

The system presented in this paper has two parts: the creation of the "truster" and its application.

4.1 Building the model

Figure 1 is a block representation of the approach.

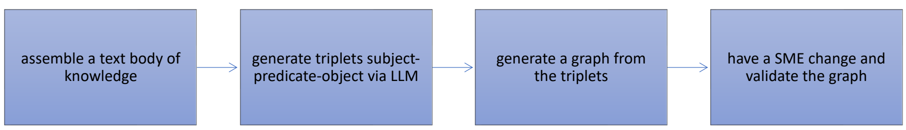

Figure 1

4.2 Collecting the body of knowledge

As mentioned above, the process of "trust" is subjective and related to specific individuals, groups, industries and cultures. In this phase, a "representative" set of text is collected.

Being the whole system rooted in the same Machine Learning approach as LLMs, it is subject to the same limitations: underrepresented domains provide unsatisfactory results. The more focused the knowledge domain is, the easier it is to collect text sufficiently representing it.

On the opposite side of the spectrum, if the goal is a generic "trust", meaning a common sense-like trust, we should rely on datasets comparable to those used to train the leading LLMs. This would raise the issues of feasibility - due to the computational complexity - and separation between testing and training, being the documents we would use are likely part of the training set used for the LLM we are analyzing.

4.3 Generating subject-predicate-object triplets

The goal of this phase is to represent the body of knowledge in a simple yet powerful form to model and understand information. The use of subject-predicate-object triplets is quite common in linguistics (such as in (Prasojo et al., 2018)). In this case, it also offers two benefits: making the text ready for encoding using the sentences generated by the triplets as tokens and using the triplets to create semantic graphs. Both the benefits will be detailed in the following paragraphs.

An LLM has been used to create a first draft of the triplets - details in the case study.

4.4 Generating a graph from the triplets

Up to this point, the process could have issues related to the representativeness of the text/body of knowledge and how the LLM transformed it into triplets. This is why a human subject matter expert should be involved to review and validate the results.

For as simple and powerful as the triplets' representation of the text could be, a long list of them may be difficult for a human to validate. Far easier to read and interact with is a directed graph where the subject and objects are nodes and predicates are edges.

4.5 Having an SME edit and validate the graph

The triplets and the semantic graph extracted from them may have an inaccurate domain representation. This is why a human in the loop as an SME is introduced in this phase. Using GUI-based tools, humans can edit graphs easily. Once the SME is OK with the results, the edited graph can be saved and placed back in the process.

4.6 Alternative presentation and use of the model

Besides the graph, the model could be represented as a set of sentences expressing the key elements of the system. Sentences could also be vectorized to get a numerical representation of the system. This could be used in " Retrieval-Augmented Generation (RAG)" approaches to enquiring the system, providing a way to interact in plain English with the SME-validated representation of the system. It could also be used as a second validation of the model, as described below.

Figure 2 is a block representation of the approach that include the steps described above.

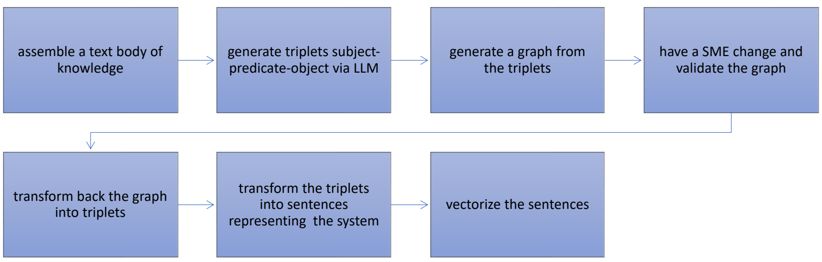

Figure 2

4.7 Transforming back the graph into triplets

The final goal is to compare a representation of the domain/body of knowledge with questions or statements from a user. The comparison could be made between text - via vectorization - or via graphs representing both sides. Comparing graphs has been done in different ways, some leveraging on the topological elements of the graph, such as in (Zhu & Iglesias, n.d.) and (Alkhamees et al., 2021), some using embeddings extracted from the graph, such as (Xiao et al., n.d.).

Because of the above, the graph has been reverted into triplets for further processing.

4.8 Transforming the triplets into sentences

For evaluating the validity of the body of knowledge, pre classified sentences - as true or false - have been used, comparing the vector representations for both. By transforming the triplets into sentences, the body of knowledge has been tokenized in a semantically relevant way. Generally speaking, the process of text tokenization for embedding generation is complex, with various challenges and potential solutions, all with limitations (Mohan et al., 2016). There are recent studies exploring new approaches, such as (W. Sun et al., n.d.), but it still remains a critical step at the foundation of machine learning approaches to language processing.

Sentences generated by triplets are tokens with semantic relevance and they could be a good foundation for embedding generation.

Once this step is completed, the body of knowledge is transformed into a set of sentences representing the given domain and vetted by a human SME. This is an evolution of a result developed by the author as "room theory" (Lipizzi et al., n.d.) before the LLMs' availability.

4.9 Vectorize sentences

The sentences are then vectorized. A transformer-based model has been used for vectorization, which is currently considered to achieve good results in this task (Rahali & Akhloufi, 2023). The resulting embeddings are stored in a vector database. The vectorized body of knowledge will act as a computational version of the knowledge base for the system to be modeled.

4.10 Second model validation

While the approach described here could be used for any query in plain English to the model representing the system, we used it to provide a second validation of the model. In order to do that, we defined two sets of sentences: one stating true facts about the system and a second stating false facts. The model should be able to discriminate them. We will use the same model used to vectorize the sentences from the model to vectorize the sentences to be analyzed. Figure 3 is a block representation of the approach.

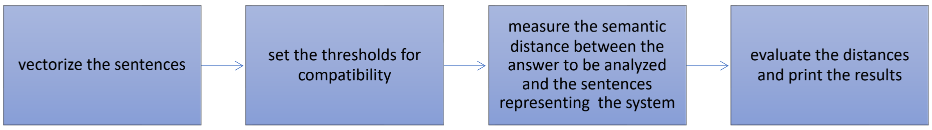

Figure 3

4.11 Vectorize the two sets of sentences

The two sets of sentences are vectorized using the same model used to vectorize the body of knowledge. Due to the limited size of the embeddings in this case, they can be handled in memory without requiring vector databases.

4.12 Set the thresholds for compatibility

This is a key step that has been addressed for this study only empirically, but it will be subject to further needed developments. Each of the sentences can or cannot be compatible with the body of knowledge. The compatibility of the individual sentences contributes then to the compatibility of the whole answer. That means that two thresholds need to be set: at the individual sentence level (t_1) and at the entire set of sentences making the answer (t_2).

4.13 Measuring the semantic distances

The measures have the scope of estimating the distance between the vectors representing the answer and the vectors representing the body of knowledge. There are several ways to measure distance between vectors (Zhao, 2009). For this study, a plain cosine similarity was used as a measurement embedded in the specific vector database. This is another critical step, and it will be the subject of future developments. For example, in similar past studies conducted by the author, variations of Word Mover's Distance (Kusner et al., 2015) provided some advantages to cosine similarity.

4.14 Evaluate the distances and print the results

Once the thresholds have been set, the system calculates the distance between each sentence and each of the sentences in the body of knowledge, printing those matching for values at or above the set threshold t_1. A summation of the individual scores creates the overall answer compatibility, which is then compared with the overall compatibility threshold t_2. The system will then print the result of the analysis in terms of compatibility or not. In case of non-overall compatibility, but in the presence of some sentence-level compatibility, a separate message will be printed.

5. Proof of concept

A simple case study has been developed to test the applicability and validity of the methodology. Being a proof of concept, it has not the breadth and depth of a real application that will be developed in a future study.

5.1 The body of knowledge

A description of "supply chain" has been used for this proof of concept. The description has been extracted from the chapter 1 of the M. Hugos book "Essentials of Supply Chain Management" (Hugos, 2024) . The chapter ("Basic Concepts of Supply Chain Management") in pdf format has been transformed in a txt file using a Python script that also preprocessed the text, applying basic NLP steps, such as the removal of non-semantically relevant words and characters. The chapter is 42 pages long, including pictures (not transformed into text).

5.2 The extraction of the triplets

GPT-4 via OpenAI API has been used to extract the triplets. This is another element that will be subject to revision, as well as testing different LLMs. The API has been used in "chat-completion" mode. In this modality, one text file has been provided with the prompt for the "assistant" and one for the "user". The "assistant" instructed the system to "extrapolate as many relationships as possible" in the form of [ENTITY 1, RELATIONSHIP, ENTITY 2], asking for a JSON format. The "user" was the text with the description of the body of knowledge related to the supply chain. The results have been preprocessed with basic NLP steps and placed in a CSV file for future reference. The system generated 360 triplets, that look like "supply chain, includes, sourcing".

5.3 Generate and validate the graph

The triplets have been used to generate the edge list that is the input for the graph representing the given body of knowledge. The process is done using Python/NetworkX and saved in GML format. The graph generated by the system has 446 nodes (representing the unique entities) and 360 edges (matching the number of triplets).

The GML file has then been imported into Gephi - an Open-Source graph editor - for the SME to perform the editing/vetting.

For this proof of concept, no SME has been involved, and no structural editing has been done on the graph. Thanks to its easy user interface, making changes to the graph could be done easily by an SME with no graph technical knowledge. Figure 4 contains a representation of the graph.

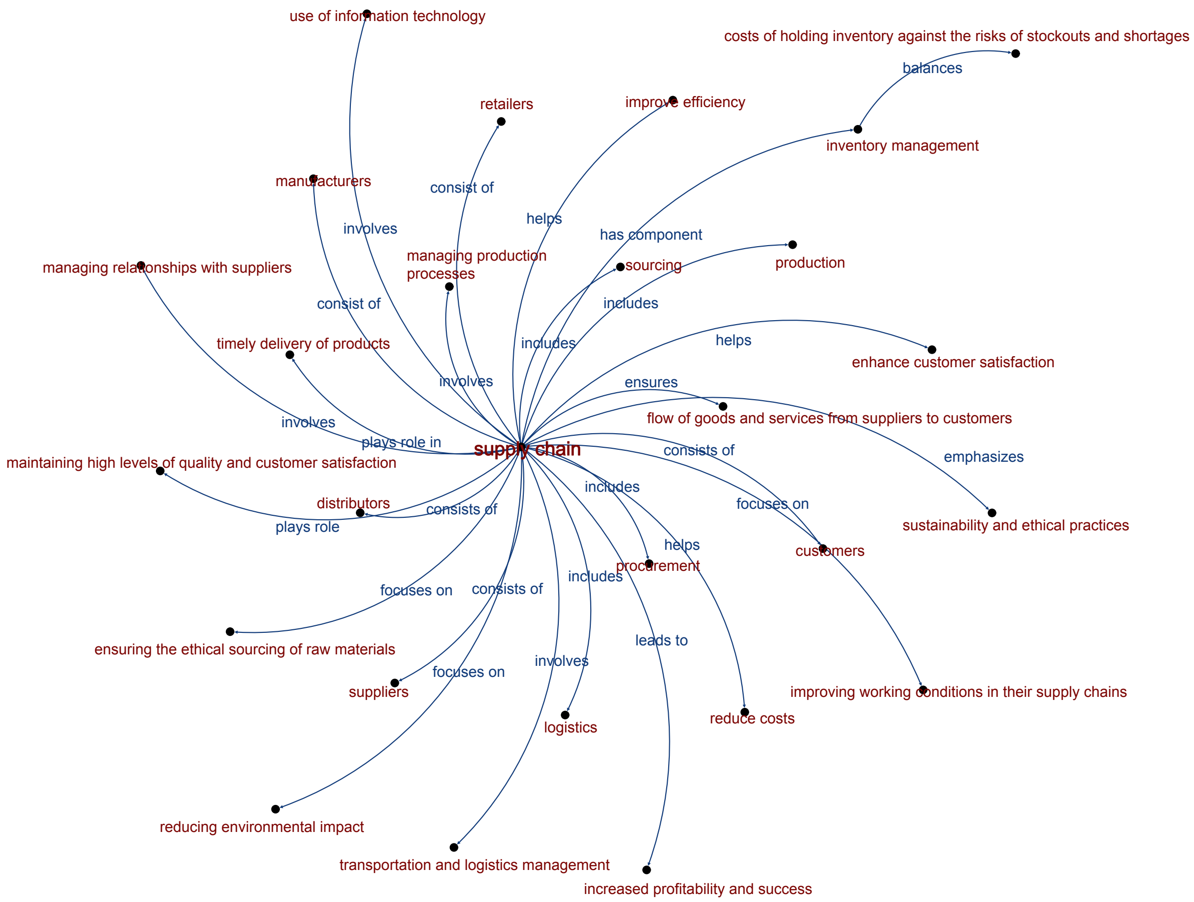

Figure 4

5.4 Transforming the triplets into sentences and vectorize them

Because no change has been made to the graph, the same triplets that are creating the graph have been used for the rest of the pipeline. If this would not have been the case, a further step graph->triplets would have been required. The transformation of triplets to sentences has been done in Python, generating sentences like "supply chain includes sourcing". A total of 360 sentences has been generated, matching the number of triplets.

For this proof of concept, the sentences have been vectorized using the all-roberta-large-v1 model, with the vectors then placed in a Pinecone vector database.

5.5 Placing questions and getting answers

A set of 55 test queries was created to evaluate the system's ability to distinguish between meaningful and nonsensical inputs. These queries include both plausible sentences - statements that are logically and contextually appropriate within a business or supply chain domain (e.g., "Inventory management involves balancing costs and risk") - and non-plausible sentences -statements that are grammatically correct but semantically nonsensical or irrelevant in context (e.g., "Supply chain plays football").

5.6 Generating triplets from the queries, transforming the triplets into sentences and vectorize them

The sentences have been vectorized using the same all-roberta-large-v1 model and placed in an in-memory storage.

5.7 Set the thresholds for compatibility

The threshold (t_1) has been empirically set for individual sentence level. As mentioned in the methodology, this step will be subject to further developments.

5.8 Measure and evaluate the distances/ print the results

Pinecone's embedded cosine similarity has been used to measure the distance between the "queries" and the sentences in the body of knowledge. The following is an example of a compatible answer:

---Query: suppliers provide materials

Matched sentence: supply chain includes sourcing

- score: 0.65

Matched sentence: supply chain includes procurement

- score: 0.62

Matched sentence: supply chain consists of suppliers

- score: 0.6

---> The semantic proximity of this phrase to the knowledge base is: 1.88

That means the phrase is compatible with the knowledge base

The following is a sample of partial compatibility:

---Query: suppliers provide money

Matched sentence: supply chain includes sourcing

- score: 0.64

Matched sentence: supply chain consists of suppliers

- score: 0.63

---> The semantic proximity of this phrase to the knowledge base is: 1.27

That means the phrase is not compatible with the knowledge base

but there is some minimal compatibility

This is a sample of non-compatibility:

---Query: supply chain plays football

-- No match: the phrase is not compatible with the knowledge base

The system demonstrated a consistent ability to distinguish between meaningful and irrelevant inputs: for example, the plausible query “suppliers provide materials” achieved a high cumulative similarity score (1.88), while nonsensical input like “supply chain plays football” yielded no significant match. Queries of partial relevance, such as “suppliers provide money”, received intermediate scores, reflecting limited semantic alignment. These results indicate that the model can effectively assess compatibility with domain knowledge, although threshold calibration remains an area for future refinement.

6. Conclusions and Future Work

This work presents a comprehensive and systematic approach to building trustworthy models of complex systems by leveraging the capabilities of large language models (LLMs) and structured semantic representations. By starting with the careful collection of domain-specific textual documentation - such as detailed system requirements - , the methodology translates unstructured knowledge into structured formats — subject–predicate–object triplets — that are further validated and enriched through expert review. These triplets form the foundation of a semantic graph that models the system's entities and their interrelations. This human-in-the-loop validation ensures that the generated model reflects a reliable and contextually accurate representation of the system. The result is not just a static model but a validated, machine-readable structure that can be queried, adapted and reused, offering a dynamic representation of system knowledge.

Beyond model creation and representation, this approach incorporates a query system in plain English to the model itself. This approach has been used in this paper as a secondary validation pipeline where queries about the system - pre-classified as true/false - are evaluated against the verified knowledge base. By transforming both the knowledge model and queries into vectorized sentence embeddings, the system can assess semantic compatibility using similarity metrics. This enables a quantitative check on how well LLM outputs align with the validated system model, reinforcing the system's ability to maintain structural fidelity and resist the inclusion of hallucinated or domain-inconsistent information. As demonstrated in the proof of concept, this framework provides a practical path toward developing LLM-augmented tools that are not only powerful but also dependable and transparent, laying the groundwork for future applications in engineering, compliance and decision support systems.

We are exploring different avenues for both the formal representation and validation. In particular, since SysML (Systems Modeling Language) is one of the most common standards in systems engineering, SysML models could be a replacement for the plain graphs used in this paper. For the validation part, mathematical formal validation approaches are also being explored.

References

Alkhamees, M.A., Alnuem, M.A., Al-Saleem, S.M., Al-Ssulami, A.M. (2021). A semantic metric for concepts similarity in knowledge graphs. Journal of Information Science, 49(3), 778–791. DOI: 10.1177/09610006211029243

Amabile, T.M. (1996a). Creativity in Context. Westview Press, New York. Book - DOI n.a.

Amabile, T.M. (1996b). Entrepreneurial creativity through motivational synergy. Journal of Creativity Behavior, 31, 18-26. DOI: 10.1002/j.2162-6057.1997.tb00766.x

Barile, J., Margolis, A., Cason, G., Kim, R., Kalash, S., Tchaconas, A., Milanaik, R. (2024). Diagnostic Accuracy of a Large Language Model in Pediatric Case Studies. JAMA Pediatrics. DOI: 10.1001/jamapediatrics.2023.5750

Bliese, P.D. (2000). Within-Group Agreement, Non-Independence, and Reliability: Implications for Data Aggregation and Analysis. In: Klein, K.J., Kozlowski, S.W.J. (Eds.), Multilevel Theory, Research, and Methods in Organizations. Jossey-Bass, San Francisco, pp. 349-381. Book chapter - DOI n.a.

Boose, J.H., Bradshaw, J.M., Koszarek, J.L., Shema, D.B. (1992). Better group decisions: using knowledge acquisition techniques to build richer decision models. Proceedings of the 25th Hawaii International Conference on System Sciences. DOI: 10.1109/HICSS.1992.183296

Brown, D.J.H. (1989). A knowledge acquisition tool for decision support systems. ACM SIGART Bulletin, 108, 93–97. DOI: 10.1145/63266.63278

Clark, E., August, T., Serrano, S., Haduong, N., Gururangan, S., Smith, N.A. (n.d.). Human Evaluation of Generated Text. Retrieved March 1, 2024. DOI: 10.48550/arXiv.2107.00061

Gambetta, D. (2000). Can We Trust Trust? In: Trust: Making and Breaking Cooperative Relations. University of Oxford. Book chapter - DOI n.a.

Goodrich, B., Rao, V., Saleh, M., Brain, G., Liu, P.J. (n.d.). Assessing the Factual Accuracy of Generated Text. ACM SIGKDD Explorations. Preprint - DOI: 10.1145/3292500.3330955

Hugos, M. (2024). Essentials of Supply Chain Management (4th ed.). Wiley. Book - DOI n.a.

Jacob Menick, J., Trebacz, M., Mikulik, V., et al. (2022). Teaching language models to support answers with verified quotes. arXiv. DOI: 10.48550/arXiv.2203.11147

Koehl, D., Vangsness, L. (2023). Measuring Latent Trust Patterns in Large Language Models. Proceedings of the Human Factors and Ergonomics Society Annual Meeting, 67(1), 504–511. DOI: 10.1177/21695067231203004

Kusner, M.J., Sun, Y., Kolkin, N.I., Weinberger, K.Q. (2015). From Word Embeddings to Document Distances. International Conference on Machine Learning. DOI: 10.48550/arXiv.1511.05527

Lipizzi, C., Borrelli, D., De Oliveira Capela, F. (n.d.). The "Room Theory": a computational model to account subjectivity in NLP. DOI: 10.48550/arXiv.2005.06059

Mohan, V., Vijayarani, S., Janani, M.R. (2016). Advanced Computational Intelligence. An International Journal, 3(1). DOI n.a.

P. Bhandari, Brennan, H.M. (2023). Trustworthiness of Children Stories Generated by Large Language Models. International Conference on Natural Language Generation. DOI: 10.48550/arXiv.2308.00073

Prasojo, R.E., Kacimi, M., Nutt, W. (2018). Modeling and Summarizing News Events Using Semantic Triples. The Semantic Web: ESWC 2018, LNCS 10843, 512–527. DOI: 10.1007/978-3-319-93417-4_32

Rachith Aiyappa, J., An, J., Kwak, H., Ahn, Y.Y. (2023). Can we trust the evaluation on ChatGPT? TRUSTNLP. DOI: 10.48550/arXiv.2303.12767

Rahali, A., Akhloufi, M.A. (2023). End-to-End Transformer-Based Models in Textual-Based NLP. Applied Informatics, 4(1), 54–110. DOI: 10.3390/appliedinformatics4010004

Rejeleene, R., Xu, X., Talburt, J. (2024). Towards Trustable Language Models: Investigating Information Quality of Large Language Models. DOI: 10.48550/arXiv.2401.13086

Rosanas, J.M. (2004). Bounded Rationality and Trust. SSRN. DOI: 10.2139/ssrn.599607

Rundall, T.G., Martelli, P.F., Arroyo, L., et al. (2007). The Informed Decisions Toolbox. Journal of Healthcare Management, 52(5), 325–341. DOI: 10.1097/00115514-200709000-00008

Sun, L., Huang, Y., Wang, H., et al. (2024). TrustLLM: Trustworthiness in LLMs. GitHub. DOI n.a.

Sun, W., Yan, L., Chen, Z., et al. (n.d.). Learning to Tokenize for Generative Retrieval. DOI: 10.48550/arXiv.2304.04171

Survey: Large Language Model Adoption Reaches Tipping Point. Arize AI Blog. Retrieved February 24, 2024. DOI n.a.

Xiao, H., Huang, M., Meng, L., Zhu, X. (n.d.). SSP: Semantic Space Projection for Knowledge Graph Embedding with Text Descriptions. DOI: 10.48550/arXiv.1604.04835

Yang Liu, Y., Yao, Y., Ton, J.F., Zhang, X., Guo, R., Cheng, H., Klochkov, Y., et al. (2023). Trustworthy LLMs: A Survey and Guidelines. arXiv. DOI: 10.48550/arXiv.2308.05374

Yavuz, U., Hasiloglu, A.S., Kaya, M.D., Karcioglu, R., Ersoz, S. (2005). Developing a marketing decision model using a knowledge-based system. Knowledge-Based Systems, 18(2–3), 125–129. DOI: 10.1016/j.knosys.2004.10.007

Yost, J., Dobbins, M., Traynor, R., et al. (2014). Tools to support evidence-informed public health decision making. BMC Public Health, 14(1). DOI: 10.1186/1471-2458-14-S2-S1

Ytsen Deelstra, S., Nooteboom, H., Kohlmann, J.V.D., Innanen, S. (2003). Using knowledge for decision-making purposes. DOI 10.1016/S0195-9255(03)00070-2.

Yue Huang, Zhang, Q., Yu, P.S., Sun, L. (2023). TrustGPT: A Benchmark for Trustworthy and Responsible Large Language Models. arXiv. DOI: DOI: 10.48550/arXiv.2310.05315

Zhao, J. (2009). Vector similarity measurement method. Technical Acoustics. DOI n.a.

Zhu, G., Iglesias, C.A. (n.d.). Computing Semantic Similarity of Concepts in Knowledge Graphs. IEEE Trans. on Knowledge and Data Engineering. DOI: 10.1109/TKDE.2016.2610428

Contact: C. Lipizzi, Stevens Institute of Technology, 1 Castle Point, Hoboken 07030, United States of America, (201)-216-3303, clipizzi@stevens.edu