A Brief Overview of Key Quality Metrics for Knowledge Graph Solution Illustration on Digital NOTAMs

Juliette Mattioli[1], **Lucas Mattioli**[2], **Martin Gonzalez**[2]

[1] Thales, France;
[2] IRT SystemX, France;
juliette.mattioli@thalesgroup.com; {lucas.mattioli,martin.gonzalez}@irt-systemx.fr

Abstract

After a brief introduction of the Knowledge Graph (KG) technology, a subfield of symbolic AI used to represent and manage semantic information, this article is devoted to quality assessment, emphasizing the importance of developing trustworthy AI in such knowledge-based systems, particularly in safety-critical applications. In this context, we remind several metrics or methods that can be applied to KGs, along with examples of their implementation in the context of digital NOTAMs (**No**tice **T**o **A**ir**M**en) illustrated by the HLIF2024 Hackathon.

High-risk AI Systems Must be Qualified

The EU classifies AI systems as high-risk if they endanger citizens' health, safety or fundamental rights. Found in sectors such as transport, healthcare, defense..., these systems are designed to be reliable and secure, adhering to strict standards and undergoing rigorous processes. To improve the design and operation, it is essential that we can measure trustworthiness (Sohier et al. 2025). This involves using metrics such as precision and recall to measure accuracy and Fleiss Kappa to measure reliability. However, trustworthiness cannot be measured, as it is not a physical property. This makes it challenging to obtain reliable measures of trustworthiness due to the various stakeholders involved. Nevertheless, trustworthiness in these systems is closely linked to accountability, which can be seen as an alternative to or factor of trust (O'Neill 2014). Dependability, encompassing safety, reliability, and maintainability, is also crucial (Avizienis et al. 2004). The Assessment List for Trustworthy AI (ALTAI 2019) outlines seven pillars of trustworthiness: human agency; technical robustness (including cyber-security (Kapusta et al. 2024)); privacy; transparency; fairness; societal welfare; and accountability.

While most academic and industrial research on trustworthy Machine Learning (ML) focuses on the properties of algorithms, holistic modeling of the quality of AI-based systems, particularly in the context of symbolic or hybrid AI, has largely been overlooked. (Kaur et al. 2022) developed an algorithm that considers 23 metrics to compute a global trustworthiness score in the context of ML. The OECD proposes a catalog of tools and metrics[1] for trustworthy AI, designed to support stakeholders in developing/using trustworthy AI solutions that respect the seven AI trustworthiness principles. Similarly, the French Confiance.ai program (Mattioli et al. 2024b) describes various attributes that contribute to the concept of trustworthiness, covering system, algorithm and data levels. This research closely identifies Key Performance Indicators (KPIs), assessment methods and checkpoints. Nevertheless, the existing literature on how to assess the quality of an AI-based system in the context of knowledge-based systems is not as prolific. (Simsek et al. 2023) identifies 23 dimensions that can be applied on KG which are Accessibility, Accuracy, Appropriate amount, Believability, Completeness, Concise representation, Consistent representation, Cost-effectiveness, Ease of manipulation, Ease of operation, Ease of understanding, Flexibility, Free-of-error, Interoperability, Objectivity, Relevancy, Reputation, Security, Timeliness, Tracability, Understandability, Value-added-ness and Variety. Indeed, the importance of context cannot be overstated, given the significant variations in trustworthiness requirements across various domains, including transportation and healthcare.

This article focuses on how to assess the quality of a knowledge-based system (Wang et al. 2021). After a brief overview of Knowledge Graph (KG) technology, we highlight the importance of evaluating its quality, a mandatory requirement for deploying critical systems. We review various metrics and methods illustrated by their implementation in the context of digital NOTAMs. Some of these were used to evaluate the various candidates in the HLIF2024 hackathon (Laudy et al. 2024), providing information on the strengths and weaknesses of the different solutions proposed. However, analysis of this hackathon, (Laudy et al. 2025) indicated that it was necessary to contextualize the qualification in relation to the queries made.

Short Introduction to Knowledge Graphs

In recent years, data-driven AI, which leverages ML techniques, has gained prominence, overshadowing symbolic AI. **Connectionist and statistical AI** operates on a biological paradigm inspired by the human brain, with the aim of derived a conceptual model from examples. This approach

[1] https://oecd.ai/en/catalogue/metrics

excels in perception but struggles with complex problem-solving. Deep learning and generative AIs, such as Large Language Models and Latent Variable Models, fall under this category. In contrast, **symbolic AI** employs formal reasoning and logic, adopting a Cartesian approach to intelligence where knowledge is encoded from axioms to deduce consequences. The key difference is that symbolic AI explicitly defines knowledge through expert input, whereas connectionist and statistical AI automatically infers knowledge from data. Therefore, in many operational contexts, particularly in critical systems operating in dynamic and uncertain environments, it is necessary to take into account information beyond raw sensor data, such as that encapsulated in physical models (simulation tools, partial differential equations, etc.) or domain knowledge captured through ontologies, logical rules, and semantic models.

A knowledge graph (KG) is a symbolic AI technology using a graphical structure to represent knowledge. It consists of nodes (entities) and edges (relations), which are used to store and represent information/knowledge related to entities and their relations (Chen et al. 2020). Their primary benefit lies in their ability to present intricate data and information in a systematic and interconnected way, which makes it easier to analyze data, retrieve information, and make decisions. In 1977, Feigenbaum (Feigenbaum et al. 1977) introduced knowledge engineering, applying AI principles for knowledge representation. KGs emerged in the 1990s within NLP, focusing on information extraction, particularly named entity recognition and relation extraction. In 1989, Berners-Lee (Berners-Lee et al. 1992) invented the World Wide Web (WWW), leading to the proposal of the Semantic Web in 1998 to integrate AI within the WWW.

KGs are directed, labeled, multi-relational graphs with semantics, representing real-world entities and their relationships. They include a reasoner or inference engine to derive implicit information from explicit concepts, unlike non-relational databases. Notable examples include DBpedia, Freebase, Wikidata, and YAGO, covering various domains and often derived from Wikipedia or created by volunteer communities (Heist et al. 2020). In 2012, Google introduced the Google Knowledge Graph, enhancing the effectiveness of its search engine by modeling and linking structured information on the internet. Thus, KGs can be formalized as finite bipartite graphs, with nodes divided into concept and conceptual relation nodes. Concept nodes represent classes of individuals, while conceptual relation nodes illustrate relationships between these concepts (Sowa 1976; Mugnier and Chein 1992). They acquire information, integrate it into an ontology, and apply a reasoner to derive new knowledge. Property graphs, where nodes and relations have properties or attributes, are commonly used (Ji et al. 2021). Entities and their relationships are represented as triples, such as $((Paris, IsCapitalOf, France)$, where "Paris" and "France" are entities, and "IsCapitalOf" is the relation.

Hence, a KG is defined as $\mathcal{KG} = \{\mathcal{E}, \mathcal{R}, \mathcal{F}\}$, where $\mathcal{E}$, $\mathcal{R}$, and $\mathcal{F}$ represent the sets of entities, relations, and facts. The fact is in a triplet format $(\varepsilon_h, \rho, \varepsilon_t) \in \mathcal{F}$, where $\varepsilon_h, \varepsilon_t \in \mathcal{E}$, and $\rho \in \mathcal{R}$, between them.

Assessment of Knowledge Graph Quality

Since high quality of data ensures its fitness for use in a wide range of applications, it is crucial to have the right metrics or methods to assess and improve the quality of a KG (Xue and Zou 2022). Based on existing work on data/information quality in general (Mattioli et al. 2022), we focus on some dimensions for assessing the quality of KG: accuracy, correctness, completeness, timeliness... Accordingly, the definition, dimensions on ML quality assessment can be transferred to knowledge graphs, as those did in (Lehmann et al. 2016). As the term quality is commonly described as "*fitness for use*" which encompasses several dimensions such as accuracy which refers to the extent to which entities and relations – encoded by nodes and edges in the graph – correctly represent real-life phenomena. Fine-grained quality measurements on single predicates or classes contribute feedback to the design process. They also allow further correction. Nevertheless, this issue has thus far escaped the attention of academic research. The main method of conducting a quality assessment is usually a manual evaluation.

In what follows, $|r|$ denotes the norm of a graph r, that is, its number of edges.

Quality Assessment Process

The KG qualification process (Mattioli et al. 2024a; Awadid et al. 2024) consists of the following activities:

1. *Characterize trustworthiness dimension*. This activity involves thoroughly organizing and detailing all the characteristics derived from the requirements of the specified attribute (e.g., correctness, completeness, consistency...).
2. *Implement metrics or methods*. Next, from the description of attribute, metrics or methods for each characteristic are implemented.
3. *Assess and report*. During development, or later at deployment time, the implemented metrics/methods are applied on a specific attribute. For instance, the accuracy characteristic reflects the ground truth of knowledge item, or correctness verifies that all the knowledge items are correctly represented. The result, available in a report, is an objective result from the implemented metrics. From this result, an improvement through KG completion and correction could be used.

The second activity mentioned above of "characterization of trustworthiness assessment" is itself broken down in several activities, according to the multi-criteria method (Mattioli et al. 2023). Those are:

1. *Define quality attributes*. All the characteristics of the considered item are identified and described (i.e., their name, properties).
2. *Structure attributes in a semantic tree*. The characteristics (i.e., quality attributes) are organized in a tree, from the most general down to the leaf characteristics.
3. *Identify numerical assessment*. Each characteristic is typed by a numerical value domain induced by the computation of the associated metrics.

4. *Adapt attribute for commensurability*. The characteristics can follow different forms of distribution, with different value domains. The purpose is to make them compatible in order to compare and operate them together.
5. *Define aggregation methodology* to explore several solutions, compare them and to keep the best one. This step is optional here, when there is a need of trade-off, not between characteristics, but between sets of characteristic values (e.g., to explore and compare sets with different levels of KPIs, more or less restrictive).

Some Quality Attributes with their Associated Metrics and Methods

Accuracy and Correctness measure the degree to which the KG reflects ground truth. Automated fact-checking tools, human expert validation, and checking facts across multiple sources can ensure the accuracy of KGs (Laudy et al. 2024). We can define KG accuracy as the percentage of triples that are correct.

The metric for correctness, μ_{correct} (Laudy and Museux 2019), ranges from 0 to 1. It calculates the information content of the result r and the the (ground truth) KG r^*, relative to r. A better result contains only the expected information. Anything extra is a negative, as it increases the operator's task. A value of 0 means the result is fully incorrect, i.e. not in r^*. A value of 1 means the result is fully correct, i.e. all the expected information is in r^*. It answers the following question: *"Did the system exactly output what was expected - and nothing more?"*

Let $r_{\text{crt}} = (r \cap r^*)|_{\Pi r}$ be the KG resulting from the projection of the intersection of r and r^* on r. The correctness is defined as:

$$\mu_{\text{correct}} = 1 - \frac{|r \setminus r_{\text{crt}}|}{|r|} \in [0, 1],$$

where $r \setminus r_{\text{crt}}$ is the complementary of r in r^*. Representing the information in r that is not present in the ground truth.

Remarks:

- A value of 1 means that all the information present in the result r, is correct (i.e. also present in the candidate answer r^*).
- A value of 0 means that none of the information present in the result r, is correct (i.e. not present in the candidate answer r^*).

Completeness measures how thoroughly the knowledge graph covers its intended domain (Issa et al. 2021). It implies that the amount of data is sufficient for the consumer's needs. This can be defined as the ratio of available data to required data, with 100% representing the optimal outcome. In real-world use cases, incomplete data can result in the loss of crucial information and consequently lead to inaccurate analysis. Furthermore, in terms of timeliness, incompleteness can compromise the ability to have all the information required at the suitable moment. It is generally challenging to evaluate the overall completeness of the knowledge graph to be tested, as there is often a lack of sufficient data to create an "ideal knowledge graph" for reference. Therefore, most researchers set their gold standard, mandatory properties, and important properties so that completeness can be assessed within a limited scope. In this way, the calculation is roughly estimated to be complete within the closed scope of human knowledge or the application domain.

The completeness (denoted as μ_{complete}) measures the quantity of r contained in r^*. The interval $[0, 1]$ is also used to define it, so that 0 means that no expected pieces of information are contained in the result provided, and 1 means that every expected piece of information is contained in the result provided. It penalizes missing relevant content. In contrast, it rewards responses that fully cover the expected answer of the question: *"Did the system retrieve all the information that was expected - and nothing less ?"*. It is the dual of correctness and follows the same principle, with r and r^* replacing each other in the formula.

Let $r_{\text{cpt}} = (r \cap r^*)|_{\Pi r^*}$ be the projection of the intersection of r and r^* on r^*. The completeness is then defined as:

$$\mu_{\text{complete}} = 1 - \frac{|r^* \setminus r_{\text{cpt}}|}{|r^*|} \in [0, 1],$$

where $r \setminus r_{\text{cpt}}$ represents the expected information that is *missing* in r, where:

- A value of 1 means that all the information present in the expected answer r^*, is also present in the result (in r).
- A value of 0 means that none of the information present in the expected answer r^*, is present in the result (in r).

Consistency and Coherence ensures the graph follows logical rules and constraints. It means that the KG is free of (logical/formal) contradictions with respect to particular knowledge representation and inference mechanisms (Lehmann et al. 2016). This attribute is widely utilized to detect incorrect knowledge in a KG (also known as error detection), only based on the provided KG itself with no additional information or external knowledge. The methods of evaluating IC are concluded as internal methods in (Paulheim 2016), which includes methods based on logic rules, knowledge embedding, statistical distribution, graph features, and so on. The study(Heyvaert et al. 2019) includes methods for checking conflicting facts, ensuring adherence to ontological constraints, and maintaining uniform representation patterns. For that purpose, it is needed to ensure that relationships follow expected patterns and that temporal information maintains chronological consistency for guarantying the integrity and reliability of the KG, ensuring that it provides a coherent and consistent view of the domain. Moreover, the concept of temporal consistency may be important for dynamic KG, where a temporal KG maintains consistency (Feng et al. 2024) among temporal relations to ensure that events with these relations create valid timelines.

Timeliness refers to how up to date the knowledge graph is in relation to the current state of the real world. In other words, a knowledge graph (KG) may be semantically accurate today, but it will become outdated if there are no procedures in place to keep it up to date. In (Rula et al. 2014), the authors were interested in providing a model for assessing timeliness and as a result they developed a metric for

currency completeness to evaluate the completeness of the timeliness measurement. This attribute is particularly important for dynamic domains where facts change frequently. Timeliness can be assessed based on how frequently the KG is updated with respect to underlying sources (Lehmann et al. 2016), which can be done using temporal annotations of changes in the knowledge graph (Rula et al. 2014), as well as contextual representations that capture the temporal validity of data. Thus, a metric associated to timeliness $\mu_{\text{timeliness}}$ can be defined as (Hartig and Zhao 2009)

$$\mu_{\text{timeliness}} = \max(0, 1 - \frac{currency}{volatility}) \quad \in [0, 1]$$

where a score of 1 implies that the data is timely and 0 means it is completely outdated and thus unacceptable. In the formula, $currency$ is the age of the data when delivered to the user and $volatility$ is the length of time the data remains valid.

Digital NOTAM Use-Case

NOTAM Definition

A NOTAM (NOtices To AirMen) is a standardized aeronautical information system used to distribute to aircraft pilots time-sensitive information about the status of airports, airspace, navigation aids, and other facilities that could affect flight safety or operations. Regular aeronautical publications don't cover temporary changes, hazards, or service disruptions, but NOTAMs act as official notifications. While aeronautical charts and airport facility directories are updated on scheduled cycles, NOTAMs can be issued immediately when conditions change to ensures that flight crews have the most accurate information needed to make decisions about flight planning, routing, and operational procedures. They are essential for aviation safety as they provide real-time updates. The system is considered unsatisfactory. Despite the increasing volume of NOTAMs, pilots and air operators still need to manually verify a lot of information.

As traditional NOTAM systems show everything, so filtering out the irrelevant information is mandatory. A survey of 2100 pilots by fixingnotam.org revealed that 72% often have difficulty understanding a NOTAM, and 74% frequently miss important information. Pilots have to read and understand many NOTAMs relevant to their flights, which is challenging and can result in accidents or near-misses due to cognitive overload. To underline this issue, a flight from Frankfurt (EDDF) to New York (KJFK). A standard briefing might show the following: 34+ NOTAMs for the departure airport; 129+ NOTAMs for the arrival airport; 100+ NOTAMs for the alternative airports and Hundreds of FIR (Flight Information Region) NOTAMs. However, if you are departing at 14:00Z, you do not need to be aware of runway closures occurring between 22:00 and 04:00.

Therefore, as NOTAM uses a specific alphanumeric format and coding system that allows pilots, air traffic controllers, and flight planners to quickly understand critical operational information (see fig. 1), NOTAM can be regarded as a semi-structured text, and its structure is neatly structured and the resource description of the spatiotemporal RDF schema model is more maneuverable. Without trustworthy digital NOTAMs, pilots would be deprived of essential real-time information. Additionally, since verifying the quality in current NOTAM messages requires manual effort, digital NOTAMs must be assess for the accuracy, completeness, and precision of their coded information.

These examples highlight the need to develop a reliable digital NOTAM analysis chain designed for automated processing and interpretation, utilising techniques such as NLP (Natural Language Processing) for entity and relation extraction, as well as symbolic AI such as knowledge graphs (KGs) to format and combine information in textual and graphical formats for human operators (Schuetz et al. 2021). Contemporary airport maps, complete with pictorial representations of ongoing developments, obstructed taxiways or runways, and transient impediments, can be provided to pilots or air traffic controllers by leveraging digital NOTAM data. Furthermore, a digital NOTAM system could automatically identify procedures affected by the unavailability of a navigation aid.

Digital NOTAM Functional Requirements

Functional requirements for a digital NOTAM system, particularly one that leverages KGs for contextualized knowledge, can be synthesized as follows:

- **Multiple Types of Entities**: The system must manage various types of entities, such as infrastructure, flight plans, airspace, and weather events.
- **Multiple Types of Relationships**: The system should support diverse relationships between entities, such as indicating the airport a runway is situated at or associating a runway with usage restrictions.
- **Schema Variability**: The system must accommodate entities of the same type having different properties and relationships. For example, runway availability can indicate inspection, closure due to snow, or restrictions based on aircraft characteristics.
- **Ontological Knowledge**: The system must handle ontological knowledge that describes relationships between classes and employs logic for defining domain-specific terms. This includes generalization/specialization relationships and domain and range constraints.
- **Domain-Specific Terms**: The system should support the use of domain-specific terminology, which is crucial for accurate representation and communication in the AIr Traffic Management domain.
- **Contextualization**: The system should use contextualized KGs to provide relevant information based on specific contexts, such as spatial, temporal, and aircraft interests.
- **Rule-Based Filtering**: The system should implement automated rule-based filtering and prioritization of NOTAMs to ensure that pilots receive the most relevant information.

By fulfilling these requirements, a NOTAM system can effectively manage and provide contextualized knowledge, offering a range of significant advantages such as:

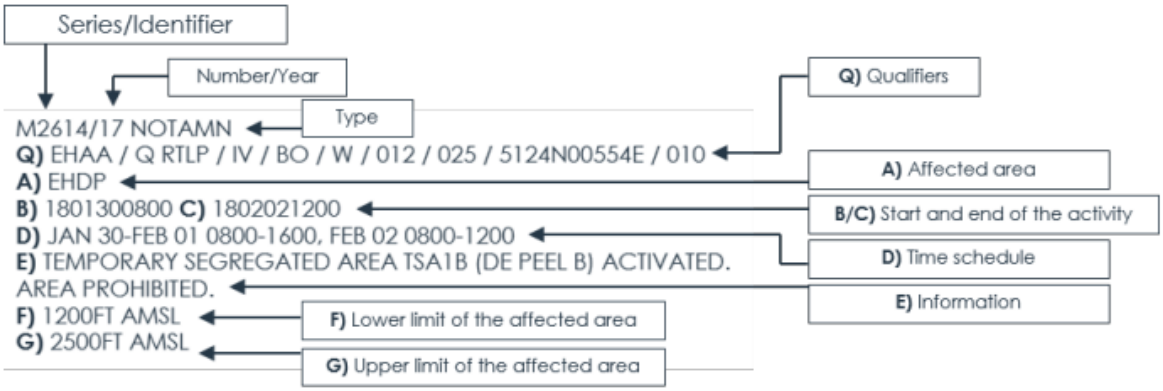

Figure 1: Example of a NOTAM.

- **Operational Efficiency**: The automation of the conversion process has been demonstrated to engender significant savings in terms of time and resources, given that it serves to reduce the necessity for manual interpretation by aviation professionals.
- The paramount importance of considerations pertaining to **safety and accuracy** cannot be overstated. Such systems offer a structured, standardized, and machine-readable format, thereby minimizing the potential for misinterpretation, which could have an impact on flight operations.
- **Scalability**: The AI system has the capacity to process large volumes of NOTAMs across different regions, thus addressing global aviation needs.

Knowledge Extraction

The first step in processing a set of raw NOTAMS is knowledge extraction, which consists of entity extraction, attribute extraction and relationship extraction. Entity extraction, which includes named-entity recognition (NER), aims to identify entities and classify them into predefined categories, such as date, airport location and runway status. The quality of entity extraction influences the efficiency and quality of knowledge acquisition, making it a fundamental process. After that, the relationships between entities are examined to establish conceptual relations. Relation extraction involves identifying relationships between entities and obtaining semantic information to construct KGs.

The quality of knowledge extraction is usually assessed using precision (also known as positive predictive value), recall (also called sensitivity), and F-measure. F-measure is the harmonic mean of precision and recall.

Completeness through Knowledge Fusion

KG completion automatically fills in missing entities or relations using knowledge fusion and reasoning techniques (Chen et al. 2020). The challenge lies in combining description information about the same entity or concept from multiple sources and integrating different KGs into a unified form. The commonly used technical methods include ontol-

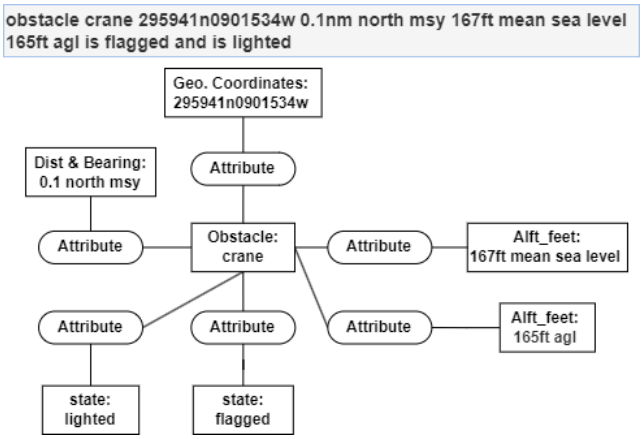

Figure 2: Knowledge extraction applied to NOTAM

ogy alignment. This is also known as ontology matching. Another commonly used method is entity alignment. This is also known as entity matching.

The goal is to realize entity alignment and domain ontology construction, which is an iterative process that includes entity alignment to assess whether or not different entities refer to the same objects in the real world. Entity alignment usually uses a variety of matching techniques combined with the features of knowledge graphs to find and match the entities that refer to the same objects. It is not simply to merge KGs, but to discover equivalent instances, equivalent attributes or equivalent classes among KGs, and to determine which entities and relationships from different KGs will be aligned. (Zhang et al. 2018) proposed to use a KG embedding and entity alignment algorithm based on representation learning for KG fusion. Another approach is introduced by (Laudy and Ganascia 2009) by using the Maximal Joint Operator of Conceptual Graph, illustrated in fig. 3 on the NOTAM use-case.

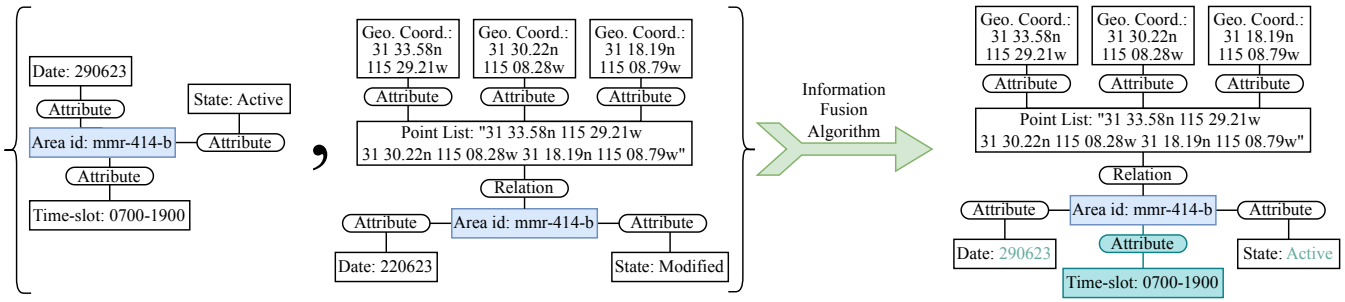

Figure 3: Knowledge fusion applied to NOTAM.

Correctness

In the context of KG, completion involves the identification of new edges or new nodes, whereas correction is concerned with the identification and removal of existing incorrect edges or nodes. The principal approaches for knowledge graph correction are divided into two main lines: fact validation, which assigns a correctness score μ_{correct} to a given fact, typically in reference to external sources; and inconsistency repairs, which aim to resolve inconsistencies found in the knowledge graph through ontological axioms.

Let us assume that the ID of the flight between Paris and Bologna is AF1328, while in KG, the same flight instance is represented as AF328 (possibly manually introduced by a data acquisition error). In this case, the node is semantically inaccurate since the flight ID does not represent its real-world state i.e. AF1328.

Consistency

If a user searches for flights between Paris and Bologna on 3 August 2025, the results are as follows:

```
Flight From To Arrival Departure
AF1328 Paris Bologna 15:30 17:05
AF1328 Paris Roma 15:30 17:05
```

The results indicate that the flight number AF1328 has two different destinations at the same date and same time of arrival and departure, which is inconsistent because one flight can only have one destination at a specific time and date. This contradiction arises due to inconsistencies in the way knowledge is represented, which can be detected using inference and reasoning.

HLIF2024 Hackathon

The HLIF2024 Hackathon[2], co-organized with the FUSION2024 conference (Laudy et al. 2024) focused on improving digital NOTAM systems. Participants were tasked to populate a KG with a given domain ontology based on NOTAM information, which could then be queried to answer the following question: "*What information in the NOTAMs is applicable to the departure and arrival airports for the period of time going from 40 minutes before the estimated time of departure to an hour and a half after the estimated time of arrival?*"

[2]HLIF stands for "High-Level Information Fusion" - HLIF2024 Hackathon website: https://fusion2025.org/hackathon/

The evaluation of participants' solutions (Laudy et al. 2025) was based on three of the key metrics described above: correctness and completeness. These metrics were chosen to assess the quality and trustworthiness of the information extracted and fused from the NOTAM data. The organizers of the HLIF2024 Hackathon provided a predefined knowledge graph (KG) that included a specific taxonomy and enforced rules for constructing IRIs for individuals belonging to the classes Airport and NOTAM. This approach allowed for the use of set intersections and subtractions rather than subgraph isomorphism searches, which are computationally intensive and time-consuming. Based on the above metrics, the HLIF2024 Hackathon organizers highlighted three approaches: AVS, F4BW, and OntoWeaver, each with its strengths and weaknesses (see table 1).

AVS aggregated the whole information from the input dataset, ensuring that all available data was included in their ontology. This approach (Chenevier et al. 2023) provides a thorough and detailed representation of the input data. AVS achieved a correctness score of 53%, indicating that a significant portion of the provided information was accurate. This suggests a robust parsing and information extraction process. AVS demonstrated a balanced performance across different metrics. The completeness score was 46%, and the overall score (average of correctness and completeness) was 49%, showing a strong mix of correctness and completeness. But, their solution resulted in a large ontology with 770 NOTAMs and 14 airports, which may not be optimal for applications requiring a small memory footprint.

F4BW pre-filtered the NOTAMs based on flight plan dates and related airports of interest, resulting in a more focused and relevant ontology. This approach is beneficial for applications where a smaller, more targeted dataset is required. F4BW achieved a correctness score of 95%, indicating a high level of accuracy in the information provided. This suggests a very precise parsing and information extraction process. Their solution resulted in a smaller ontology with 82 NOTAMs and 6 airports, which is advantageous for applications with memory constraints. Nevertheless, the completeness score was 44%, indicating that some expected information was missing from the provided ontology. This suggests that while the information is accurate, it may not be fully comprehensive.

Question	Solution	$\|r*\|$	$\|r\|$	Intersection	Difference	Symmetric Difference	Correctness	Completeness	Score
Q1	AVS	364	318	168	196	150	53%	46%	49%
Q1	FB4W	364	168	160	204	8	95%	44%	70%
Q1	OntoWeaver	364	561	142	222	419	25%	39%	32%
Q2	AVS	13	10	6	7	4	60%	46%	53%
Q2	FB4W	13	6	5	8	1	83%	38%	61%
Q2	OntoWeaver	13	21	5	8	16	24%	38%	31%
Q3	AVS	104	80	47	57	33	59%	45%	52%
Q3	FB4W	104	48	46	58	2	96%	44%	70%
Q3	OntoWeaver	104	160	40	64	120	25%	38%	32%

Table 1: Values of the metrics w.r.t. r^* from r. Score is the average of correctness and completeness (Laudy et al. 2025)

OntoWeaver proposed a generic fusion approach, targeting annotations reconciliation (Dreo et al. 2024), based on a generic software for importing table data in KGs. This approach is versatile and can handle a wide range of data fusion scenarios, including managing heterogeneous duplicates without losing information. Similar to AVS, OntoWeaver aggregated the whole information from the input dataset, ensuring a thorough representation of the input data. OntoWeaver resulted in the largest ontology with 770 NOTAMs and 14 airports, indicating a comprehensive inclusion of data. OntoWeaver achieved a correctness score of 25% and a completeness score of 39%, indicating that a significant portion of the provided information was either incorrect or incomplete. This suggests potential issues in the parsing and information extraction process.

Analyzing the Hackaton's difficulties through the lens of the End2End Methodology. Several difficulties emerged during and after the competition, largely due to underspecification of the AI system. Among them was the inability to access execution feedback without risking data leakage, typically addressed through system-level specifications. There were discrepancies between training and evaluation domains, insufficient domain expertise, and issues with unclear or inconsistent ontologies. Input data specifications evolved during the challenge; for example, robustness to duplicate NOTAMs became critical, yet was absent from the training data and should have been defined in the ODD. Output objectives and formats were ambiguous due to multi-interpretation questions, and evaluation metrics failed to account for key factors such as answer cardinality. Clearer definition of trade-offs in the AI function's output could have reduced this ambiguity.

We argue that the challenge lacked clarity on its intended purpose, automation goals, and the AI function expected of solutions before, during, and after the hackathon. The discussion section highlights the difficulty of enabling feedback without leakage, citing cases such as error messages linked to specific NOTAMs or inferring the absence of NOTAMs at certain airports. We contend these are not leaks but legitimate interactions via monitoring systems, and that handling isolated entities is itself an ODD dimension. As deduplication and isolated-node handling are standard KG challenges, their inclusion depends on the challenge's operational framing, which should specify whether the task involves inference or causal discovery. The authors also attribute divergent approaches (e.g., KGs vs. filtered graphs) to epistemological bias. We argue this stems from functional and operational under-specification. Debating the meaning of "fusion" without context is unproductive; data heterogeneity is a core challenge in Trustworthy ML, and generalization cannot be assumed across scenarios. Fusion tasks must instead clarify how they interact with heterogeneity within their operational setting. We argue that what was framed as data leakage was, in fact, a missing specification.

Our conclusion consequently differs from that in (Laudy et al. 2025): we emphasize the importance of applying the End2End methodology (Quintero et al. 2025) to frame such challenges as engineering problems, not abstract academic ones. The hackathon demonstrates this methodology's relevance not only for building AI systems but also for defining the specification dimensions that govern successful operationalization. Future work will address the factual difficulties of this challenge from an engineering perspective, remodeling its scope through the End2End methodology to establish a clear distinction between the scientific and technological problems involved in fusion within the context of NOTAM-based use cases similar to the Hackathon's.

Conclusion and Future Works

"Trust" should be integrated as a fundamental design principle rather than an optional issue, given the accelerated developments in AI. The development of AI-based systems necessitates to assess its overall quality. The reliance on accuracy for the assessment of AI-based systems can be a highly misleading exercise. Due to the multidimensional nature of trustworthiness, significant challenges include establishing objective attributes such as correctness, completeness, and precision; mapping these attributes to AI processes and their lifecycle; and developing methods and tools to assess them. Such attributes and their assessment are influenced by contextual factors, including the underlying AI technology, the application domain and the stakeholders involved. Consequently, these systems must undergo quality evaluations from various stakeholders. The evaluation process begins in the early stages of development and encompasses the definition of system specifications, analysis and design. Quality assessments must be integrated into every

stage of the system lifecycle, including specification, deployment, updates, maintenance and integration, regardless of the AI technique employed (machine learning, symbolic AI, hybrid AI or generative AI).

Acknowledgments

This work has been supported by the French government under the "France 2030" program, as part of the SystemX Technological Research Institute within the *CSIA* project.

References

ALTAI. 2019. Assessment List for Trustworthy Artificial Intelligence (ALTAI). Technical report, High-Level Expert Group on Artificial Intelligence, European Commission.

Avizienis, A.; et al. 2004. Basic concepts and taxonomy of dependable and secure computing. *IEEE transactions on dependable and secure computing*, 1(1): 11–33.

Awadid, A.; et al. 2024. Towards Engineering Processes to Guide the Development of Trustworthy ML Systems. In *2024 IEEE International Symposium on Systems Engineering (ISSE)*, 1–6. IEEE.

Berners-Lee, T.; et al. 1992. World-Wide Web: the information universe. *Internet Research*, 2(1): 52–58.

Chen, Z.; et al. 2020. Knowledge graph completion: A review. *IEEE Access*, 8: 192435–192456.

Chenevier, F.; et al. 2023. Method for processing aeronautical information intended for flight crews, FR Patent EP2 023 081 570.

Dreo, J.; et al. 2024. Reproducible Mapping of Tabular Data into Semantic Knowledge Graphs with OntoWeaver and BioCypher. In *2024 27th International Conference on Information Fusion (FUSION)*, 1–8. IEEE.

Feigenbaum, E.; et al. 1977. The art of artificial intelligence: Themes and case studies of knowledge engineering.

Feng, S.; et al. 2024. Temporal knowledge graph reasoning based on entity relationship similarity perception. *Electronics*, 13(12): 2417.

Hartig, O.; and Zhao, J. 2009. Using Web Data Provenance for Quality Assessment. *SWPM*, 526.

Heist, N.; et al. 2020. Knowledge graphs on the web–an overview. *Knowledge Graphs for eXplainable AI: Foundations, Applications and Challenges*, 3–22.

Heyvaert, P.; et al. 2019. Rule-driven inconsistency resolution for knowledge graph generation rules. *Semantic Web*, 10(6): 1071–1086.

Issa, S.; et al. 2021. Knowledge graph completeness: A systematic literature review. *IEEE Access*, 9: 31322–31339.

Ji, S.; et al. 2021. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE transactions on neural networks and learning systems*, 33(2): 494–514.

Kapusta, K.; et al. 2024. Protecting ownership rights of ML models using watermarking in the light of adversarial attacks. *AI and Ethics*, 4(1): 95–103.

Kaur, D.; et al. 2022. Trustworthy artificial intelligence: a review. *ACM computing surveys (CSUR)*, 55(2): 1–38.

Laudy, C.; and Ganascia, J. 2009. Introducing semantic knowledge in high-level fusion. In *MILCOM 2009-2009 IEEE military communications conference*, 1–7. IEEE.

Laudy, C.; and Museux, N. 2019. How to evaluate high level fusion algorithms? In *2019 22th International Conference on Information Fusion (FUSION)*, 1–8. IEEE.

Laudy, C.; et al. 2024. HLIF2024: a Competition for High-Level Information Fusion. In *2024 27th International Conference on Information Fusion (FUSION)*, 1–8. IEEE.

Laudy, C.; et al. 2025. First High-Level Information Fusion Competition: Feedback and Lessons Learned. In *28th Inernational Conference on Information Fusion*.

Lehmann, J.; et al. 2016. Quality assessment for Linked Data: A Survey. *Semantic Web (1570-0844)*, 7(1).

Mattioli, J.; et al. 2022. Information Quality: the cornerstone for AI-based Industry 4.0. *Procedia Computer Science*, 201: 453–460.

Mattioli, J.; et al. 2023. Towards a holistic approach for AI trustworthiness assessment based upon aids for multi-criteria aggregation. In *SafeAI 2023-The AAAI's Workshop on Artificial Intelligence Safety*, volume 3381.

Mattioli, J.; et al. 2024a. Leveraging Knowledge Graph to design the Machine-Learning Engineering Body-of-Knowledge. In *2024 Conference on AI, Science, Engineering, and Technology (AIxSET)*, 258–265. IEEE.

Mattioli, J.; et al. 2024b. An overview of key trustworthiness attributes and KPIs for trusted ML-based systems engineering. *AI and Ethics*, 4(1): 15–25.

Mugnier, M.; and Chein, M. 1992. Conceptual graphs: Fundamental notions. *Revue d'intelligence artificielle*, 6(4): 365–406.

O'Neill, O. 2014. Trust, Trustworthiness, and Accountability. In Morris, N.; and Vines, D., eds., *Capital Failure: Rebuilding Trust in Financial Services*, 0. Oxford University Press. ISBN 978-0-19-871222-0.

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

Quintero, K.; et al. 2025. An end-to-end method for operationalizing trustwothiness in AI-based critical systems. In *15th International Conference on Performance, Safety and Robustness in Complex Systems and Applications PESARO 2025*.

Rula, A.; et al. 2014. Capturing the currency of dbpedia descriptions and get insight into their validity. In *CEUR Workshop Proceedings*, volume 1264.

Schuetz, C.; et al. 2021. Knowledge graph OLAP: A multidimensional model and query operations for contextualized knowledge graphs. *Semantic Web*, 12(4): 649–683.

Simsek, U.; et al. 2023. A knowledge graph perspective on knowledge engineering. *SN Computer Science*, 4(1): 16.

Sohier, H.; et al. 2025. The Engineering of AI Evaluation and Scoring: Overview and Insights. In *2025 IEEE International Systems Conference (SysCon)*, 1–8. IEEE.

Sowa, J. 1976. Conceptual graphs for a data base interface. *IBM Journal of Research and Development*, 20(4): 336–357.

Wang, X.; et al. 2021. Knowledge graph quality control: A survey. *Fundamental Research*, 1(5): 607–626.

Xue, B.; and Zou, L. 2022. Knowledge graph quality management: a comprehensive survey. *IEEE Transactions on Knowledge and Data Engineering*, 35(5): 4969–4988.

Zhang, Y.; et al. 2018. Entity alignment across knowledge graphs based on representative relations selection. In *2018 5th International Conference on Systems and Informatics (ICSAI)*, 1056–1061. IEEE.