Modeling an Hybrid System (predicate-logic-based + ML/AI) for Verifying the Credibility of Information Sources

DOMINIQUE S. LOYER[1]

[UQAM] *PhD candidate in AI & assistant researcher*
[@] *loyer.dominique@uqam.ca*

May 17, 2025

Faced with growing information overload ("infobesity"), distinguishing reliable information has become a major challenge for digital citizens. The polarization of opinions, often fueled by the rapid dissemination of unverified or even false information on digital platforms, exacerbates this complex problem [39]. This project aims to design a system capable of assessing the credibility of information and its sources (websites, articles, responses from large language models). The objective is to provide the user with clear and interpretable metrics (credibility score, source analysis, identification of divergent opinions) to help them develop critical thinking and make informed decisions. The envisioned system is based on a hybrid approach, combining predefined predicate logic rules and machine learning techniques (natural language processing, sentiment analysis, coherence analysis, bias detection) to analyze content and evaluate its sources [2]. This approach is motivated by the recognized limitations of fully automated systems, particularly in complex domains where human judgment remains crucial but difficult to scale [14, 26, 30, 38]. This report presents the initial UML modeling of this system, detailing its envisioned static structure and dynamic behavior.

1. INTRODUCTION & PROBLEM STATEMENT

A. The Challenge of the Modern Information Ecosystem

The digital age has generated an unprecedented volume of information, creating a situation often described as "infobesity." Navigating this constant flow and discerning reliable information from misleading content has become an essential but difficult skill for the average citizen to master. The challenge is amplified by the speed at which information, whether true or false, spreads on social networks and other digital platforms [5, 6, 20]. This rapid dissemination far exceeds the capabilities of traditional manual fact-checking methods, which, although essential, cannot handle the scale and velocity of the problem [14, 20, 30].

At the heart of this challenge lies the proliferation of disinformation, misinformation, and "fake news" [20, 37, 39, 42]. It is useful to distinguish these terms: misinformation refers to the spread of false information without intent to harm, while disinformation involves a deliberate intent to deceive [30, 42]. "Fake news," on the other hand, often mimics the format of journalistic content but lacks rigorous editorial processes and can serve malicious purposes [37, 42]. The impact of such erroneous content is profound, affecting public opinion, democratic processes, trust in institutions, and even public health, as demonstrated by the COVID-19 pandemic [8, 20, 39, 42].

The advent of Generative Artificial Intelligence (GenAI) has added a significant layer of complexity. Technologies like large language models (LLMs) and generative adversarial networks (GANs) now allow for the automatic creation of strikingly realistic texts, images, and videos (deepfakes), often indistinguishable from authentic content by the human eye or even by traditional detection tools [6, 20, 38]. This capacity for large-scale, low-cost production of false but credible content exacerbates the information crisis [20, 37, 42].

Concurrently, users often face a lack of transparency regarding information sources. Assessing the credibility of a website, author, or platform requires time and expertise that most people do not possess [23, 30, 33, 39]. Credibility itself is a multidimensional concept, encompassing aspects such as trustworthiness, perceived expertise, factual accuracy, objectivity (or lack of bias), presentation quality, and timeliness of information [30, 33, 39]. This complexity makes evaluation all the more challenging for the non-specialist.

B. Proposed Solution: An Automated Credibility Assessment System

To address these challenges, this project proposes the design of an information credibility assessment system. The main objective is not to replace human judgment but to provide users with tools to navigate the complex information ecosystem more effectively. The system aims to generate clear and interpretable metrics – such as an overall credibility score, a detailed analysis of cited or underlying sources, and the identification of any documented diverging viewpoints – to support and encourage the development of critical thinking in the user.

A central feature of the envisioned system is its *hybrid* approach [2]. Recognizing the limitations of purely algorithmic approaches, particularly when dealing with the subtlety and contextuality of much information [14, 26, 30, 38], the system will combine:

- **Predefined predicate logic rules**[1]**:** These allow for the encoding of explicit knowledge about indicators of credibility or non-credibility (e.g., the known reputation of a source, the presence of certain linguistic markers typical of disinformation) [16, 23, 40]. They are effective for quick checks and known patterns.
- **Artificial Intelligence (AI) and Natural Language Processing (NLP) techniques**[2]**:** These are employed to analyze more nuanced aspects of the content, such as the sentiment conveyed, the semantic coherence of the text, the presence of subtle biases, and to adapt to new disinformation tactics that evade fixed rules [2]. AI plays an essential role here, not only as a potential source of the problem (via GenAI creating fakes) but also as an integral part of the solution [20, 42].

This combination aims to leverage the robustness and transparency of rules for clear cases and the flexibility of AI for complex analyses and adaptation.

The specific objectives of the system, as defined in the initial problem statement, include:

- Combatting information overload (Infobesity).
- Assisting in the detection of disinformation and "Fake News," including those generated by AI.
- Offering a transparent assessment of source credibility.
- Presenting evaluation metrics in a simple and understandable manner.
- Encouraging equipped critical thinking by presenting diverse viewpoints.
- Attempting to ensure information traceability.
- Contributing (in the long term) to the transparency and accountability of information systems.

To achieve these objectives, the system will necessarily rely on external data sources, such as search engines, LLM APIs, and verified fact databases [39]. Modeling these interactions is a key part of the work presented in this report.

C. Report Structure

This report details the first phase of the system's design: modeling using the Unified Modeling Language (UML). Section 2 presents the various UML diagrams developed: actor specification, use cases, class diagram, sequence diagrams, and state-transition diagram. Section 3 elaborates on the key technologies and components envisioned, including the rule engine, the AI/NLP module, and the integration of external data. Section 4 discusses the modeling choices, the inherent challenges in credibility assessment, the limitations of the current model, and future prospects. Finally, bibliographic references are provided, followed by the table of contents.

2. SYSTEM MODELING (UML)

The Unified Modeling Language (UML) was chosen for this design phase due to its standardized nature and its ability to represent different aspects of a complex software system, from its users to its internal structure and dynamic behavior [9, 18]. The following diagrams provide a blueprint for the development of the credibility assessment system.

[1] See Appendix II
[2] See Appendix I

A. Actor Specification

Actors represent external entities (human or systems) that interact with the assessment system. Clearly identifying actors and their goals is fundamental to defining the system's boundaries and functionalities. The following actors have been identified:

Table 1. Actors of the Credibility Assessment System

Actor	Description	Main Goals
User	(Average) person wishing to verify the credibility of information (text, URL, query).	Submit a verification request, Consult the simplified assessment report, Provide feedback on the report's relevance.
Expert	Qualified person (e.g., data scientist, fact-checker) responsible for the fine-tuning of the system.	Define/Adjust verification rules[3], Configure credibility indicators, Analyze logs to improve the system.
External System	Third-party data source (Search engine API, LLM API, Fact-checking database, etc.).	Provide raw data (articles, search results, web content) for analysis, Provide metadata (date, author, etc.).
System	The credibility assessment system itself.	Process requests, Query external systems, Apply rules and AI models[4], Calculate scores, Generate reports.

The inclusion of the "System" actor is a useful modeling convention for visualizing internal responsibilities when describing dynamic interactions (e.g., in sequence diagrams), although it does not represent an external entity in the strict sense. The "Expert" actor is crucial as it embodies the need for continuous maintenance and adaptation of the system, a fundamental requirement given the constant evolution of disinformation techniques and the inherent limitations of AI models [20, 26, 42].

B. Use Case Modeling

Use cases describe the functionalities offered by the system from the actors' perspective. They define how actors interact with the system to achieve their goals.

B.1. Use Case Diagram

The following diagram (Figure 1) illustrates the main use cases and their relationships with the actors. It highlights the central interaction of the User to submit a request and receive a report, the Expert's role in configuration, and the dependency on External Systems for data collection. The "Utilize hybrid system" use case encapsulates the internal logic combining rules and AI.

The logical groupings (implicit packages) are:

- **User Interaction:** Submit request (UC1), Generate report (UC2), Provide feedback (UC7).
- **Internal Processing:** Utilize hybrid system (UC_Hybrid), Apply rules (UC4), Analyze via AI (UC5), Calculate score (UC6). These

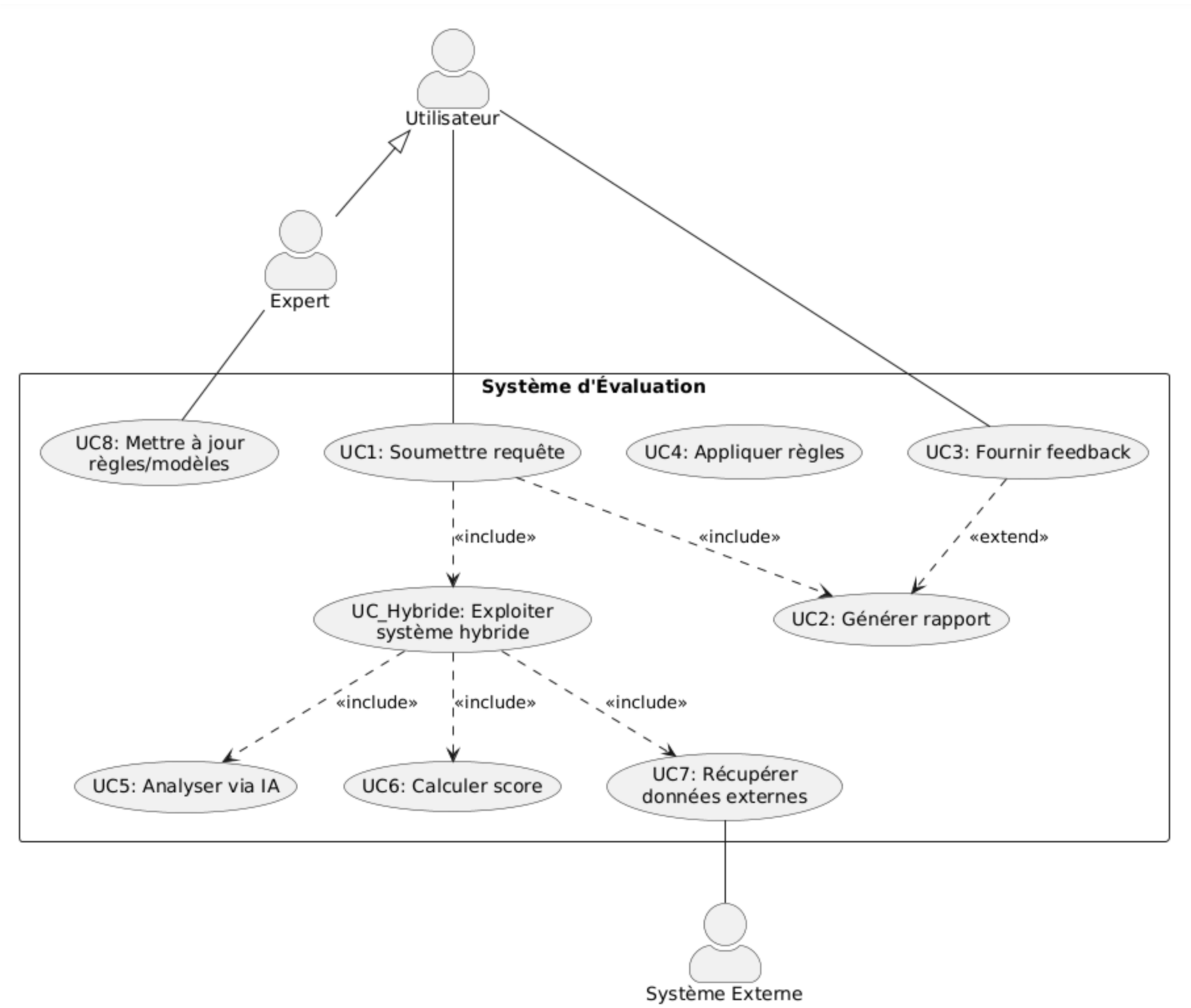

Fig. 1. Use case diagram. (Simplified)

cases represent the functional core of the system, orchestrating the hybrid analysis.

- **External Management:** Retrieve external data (UC3). This case underscores the critical dependency on third-party APIs.
- **Expert Configuration:** Update rules/models (UC8). This case ensures the system's adaptability and maintenance.

B.2. Use Case Specification

The following specifications detail the most important use cases from the perspective of external interactions.

Use Case 1: Submit a verification request (UC1)

- **Goal:** Allow the user to request an assessment of information credibility.[5]
- **Primary Actor(s):** User.
- **Secondary Actor(s):** System, External System.
- **Preconditions:** The user has access to the interface. The system is operational. External systems are (at least partially) accessible.
- **Flow:**
  1. The User enters/pastes the information (text, URL, query).
  2. The User submits the request.
  3. The System receives the request.
  4. The System triggers UC_Hybrid to process the request.
  5. The System receives the result (including the report from UC2).
  6. The System presents the report to the User (via UC2).
- **Postconditions:** Report generated and presented. Request potentially stored (anonymized).
- **Extensions:** Invalid information (user notification). External system unavailable (partial processing, indicated in the report). Timeout (proposal for deferred delivery).

Use Case 2: Generate a credibility report (UC2)

- **Goal:** Present the assessment results clearly and concisely.
- **Primary Actor(s):** System.
- **Secondary Actor(s):** User.
- **Preconditions:** UC6 (Calculate score) completed successfully. Metrics and analyses available.
- **Flow:**
  1. The System collects the results (score, rule/NLP details, sources, opinions).
  2. The System formats them into a structured and readable report.
  3. The System includes simple explanations for the metrics.
  4. The System presents the report to the User.
- **Postconditions:** Report displayed.
- **Extensions:** Variable level of detail (user choice: summary vs. detailed).

[5] See the interface (in french only as in May 2025 in Appendix 0

Use Case 3: Retrieve external data (UC3)

- **Goal:** Collect relevant information from external sources.
- **Primary Actor(s):** System.
- **Secondary Actor(s):** External System (Search Engine API, LLM API, Fact-checking API, etc.).
- **Preconditions:** UC1 submitted. Information to search identified. External APIs configured.
- **Flow:**
  1. The System identifies the necessary data (URL content, search results, similar articles...).
  2. The System sends requests to the appropriate External Systems [12, 13, 24, 35].
  3. The External Systems return the data (HTML, text, JSON...).
  4. The System preprocesses and temporarily stores the raw data.
- **Postconditions:** External data available for UC4 and UC5.
- **Extensions:** API/Scraping error (error handling, alternatives). API rate limits (respecting quotas, handling 429 errors) [12, 24]. Unreliable or inaccessible source.

The robustness of this use case is critical, as any failure in retrieving external data directly impacts the quality of the assessment. Error handling, timeouts, and API limit management must be implemented carefully [12, 24].

Use Case 8: Update rules/models (UC8)

- **Goal:** Allow an expert to improve the system's performance and relevance.
- **Primary Actor(s):** Expert.
- **Secondary Actor(s):** System.
- **Preconditions:** Expert authenticated with sufficient rights. Need for update identified.
- **Flow:**
  1. The Expert accesses the administration interface.
  2. The Expert modifies/adds/deletes rules (affecting UC4) or updates/retrains AI models (affecting UC5).
  3. The Expert tests the modifications (potentially on a validation set).
  4. The Expert deploys the modifications.
- **Postconditions:** Rules or models updated. Performance potentially improved.
- **Extensions:** Rollback to a previous version in case of problems.

This use case is essential for the system's maintainability and evolution in the face of new disinformation threats and advancements in AI techniques [20, 26, 42].

The other use cases (UC4, UC5, UC6, UC_Hybrid) are primarily internal steps triggered by UC1. Their logic is as follows: UC4 applies the logical rules defined by the Expert; UC5 executes AI-based analyses (sentiment, coherence, bias, etc.); UC6 combines the results from UC4, UC5, and source analysis (from UC3) to calculate an overall credibility score and detailed metrics; UC_Hybrid orchestrates this entire internal process.

C. Static Modeling (Class Diagram)

The class diagram (Figure 2) defines the static structure of the system: the main classes, their attributes, their methods, and the relationships between them (associations, inheritance, dependencies). It provides a blueprint for implementation.

Key points of the class diagram:

- **Central Orchestration:** The 'EvaluationSystem' class acts as a facade or main controller, encapsulating the logic to initiate an assessment. It holds references to the necessary rules ('VerificationRule'), AI models ('AIModel'), evaluation metrics ('EvaluationMetric'), and external systems ('ExternalSystem').

- **Request Management:** 'EvaluationRequest' represents a specific assessment request, linked to an 'InputInformation' and producing an 'EvaluationReport'. It also manages the processing status.

- **Extensibility of External Sources:** The use of an abstract class 'ExternalSystem' with concrete subclasses ('SearchEngine', 'ApiLLM', 'FactDatabase') allows for easy addition of new data sources without radically changing the system's core [9]. Each external source implements the 'retrieveData' method.

- **Explicit Hybrid Approach:** The distinct presence of 'VerificationRule' and 'AIModel' classes, both used by 'EvaluationSystem', directly reflects the hybrid approach. How their results ('RuleResult', 'NLPResult') are combined in the 'calculateScore' method remains an implementation detail, but the structure supports this combination.

- **Richness of the Report:** The 'EvaluationReport' class is designed to be comprehensive, including not only an overall score ('credibilityScore') but also details on sources ('SourceInfo'), applied rules ('RuleResult'), and NLP analysis ('NLPResult'). This allows for flexible presentation to the user, from summary to full detail.

- **Configuration by the Expert:** The relationships between 'Expert' and the 'VerificationRule', 'AIModel', 'EvaluationMetric' classes show how the expert interacts with the system to configure and improve it, in accordance with UC8.

This static model provides a solid structure but intentionally leaves some implementation details open, such as the exact internal logic of the rules, the architecture of the AI models, or the precise score calculation algorithm.

D. Dynamic Modeling

Dynamic modeling illustrates how objects (instances of classes) interact over time to realize the use cases. It complements the static view by showing the system's behavior.

D.1. Sequence Diagrams

Sequence diagrams describe the chronological interactions between objects for specific scenarios.

Scenario 1: Simple URL verification by the User This scenario (Figure 3) shows the typical flow when a user submits a URL for assessment. It illustrates orchestration by 'SysFacade' (representing 'EvaluationSystem'), creation of a request ('ReqMgr'), call to the hybrid engine ('HybridEng'), data retrieval via 'ExtData' from external systems ('SE1', 'SE2'), internal application of rules and AI, score calculation, report generation ('ReportGen'), and finally presentation of the report to the user. This diagram highlights the collaboration between multiple components and the crucial interaction with external APIs [9, 18]. Potential error handling during external calls (e.g., to 'SE1' or 'SE2') is not explicitly shown here but is an essential consideration for implementation, as mentioned in UC3 [12, 24].

Scenario 2: Updating a rule by the Expert This scenario (Figure 4) illustrates the process of updating a verification rule by an expert (UC8). It shows interaction via an administration interface ('AdminUI'), rule validation by a manager ('RuleMgr'), an optional but recommended testing step ('TestSys') on a dataset, and finally the effective update and saving of the rule if tests are successful. This process underscores the importance of a validation cycle before deploying new rules or AI models to maintain the system's quality and reliability.

D.2. State/Transition Diagram

The state-transition diagram (Figure 5) models the lifecycle of an 'EvaluationRequest' object, showing the different states a request goes through from submission to completion (or failure), as well as the events or actions that trigger transitions between these states [9].

The main states are:

- **Submitted:** The request is created but not processed.

- **Processing:** Retrieving external data, applying rules, and AI analyses are in progress. This state may involve internal loops or sub-states not detailed here.

- **Analyzed:** Analyses are complete, score calculation is imminent or finished.

- **ReportGenerated:** The report is ready for the user.

- **WithFeedback:** The user has provided feedback (via UC7).

- **Error:** An unrecoverable error occurred (e.g., critical data retrieval failure, major internal error).

This diagram helps to understand the overall control flow for processing a request and to identify points where errors can occur.

3. KEY SYSTEM COMPONENTS AND TECHNOLOGIES

UML modeling provides an architectural overview. This section further details the critical internal components and the technologies envisioned for their implementation, building on the modeled classes and interactions.

A. Rule-Based Engine

The component responsible for applying rules ('VerificationRule' in the UML model) plays a crucial role in the hybrid approach. Its objective is to quickly identify indicators of credibility or non-credibility based on heuristics and established knowledge, often before launching more resource-intensive AI analyses.

Concrete examples of rules could include [16, 23, 28, 30, 33, 40]:

- **Source-based rules:**
  - Checking the URL/domain against lists of sources known for their reliability or unreliability (e.g., inspired by Media Bias/Fact Check or NewsGuard assessments [30]).
  - Analyzing domain age (a very recent domain may be suspicious).
  - Checking for the presence of clear contact information or an editorial policy on the source site.
  - Analyzing the source's history of publishing false information (if data is available, e.g., via APIs like Google Fact Check API which can return 'ClaimReview' associated with specific URLs [13, 35]).

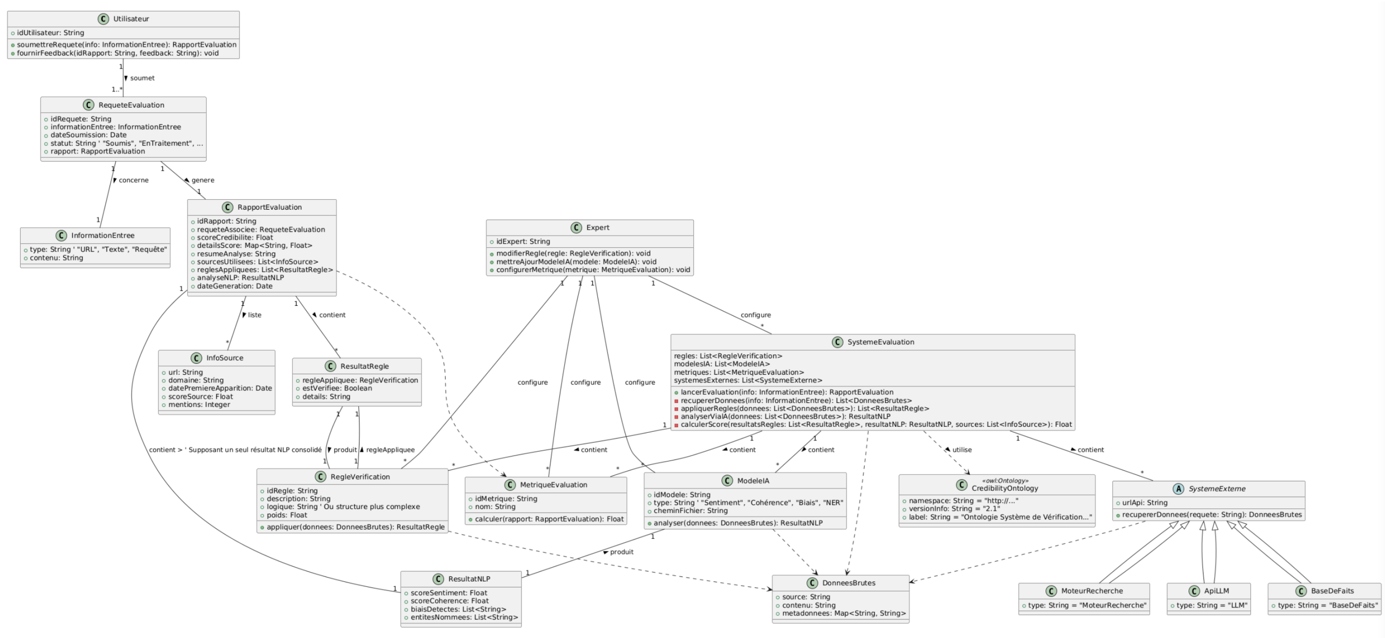

Fig. 2. Class diagram. (Class for the ontology added following comments during the presentation)

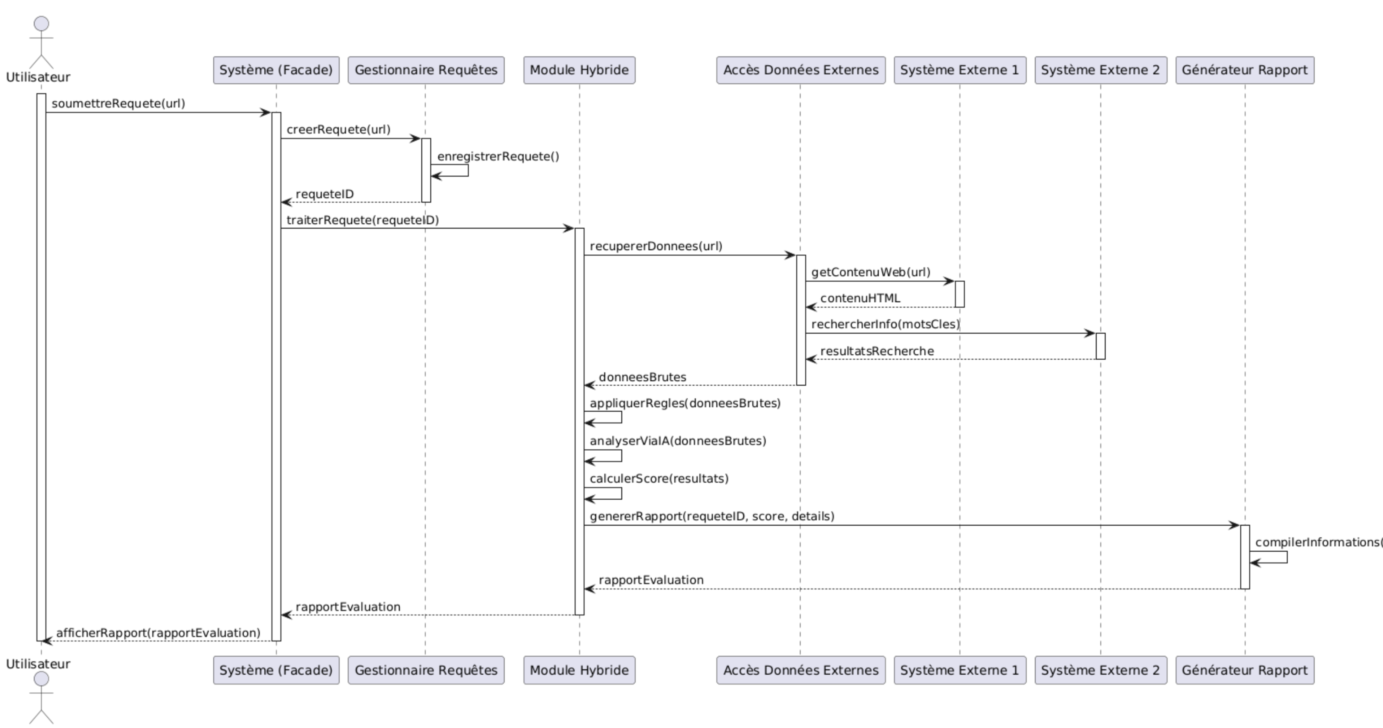

Fig. 3. Sequence diagram for scenario 1.

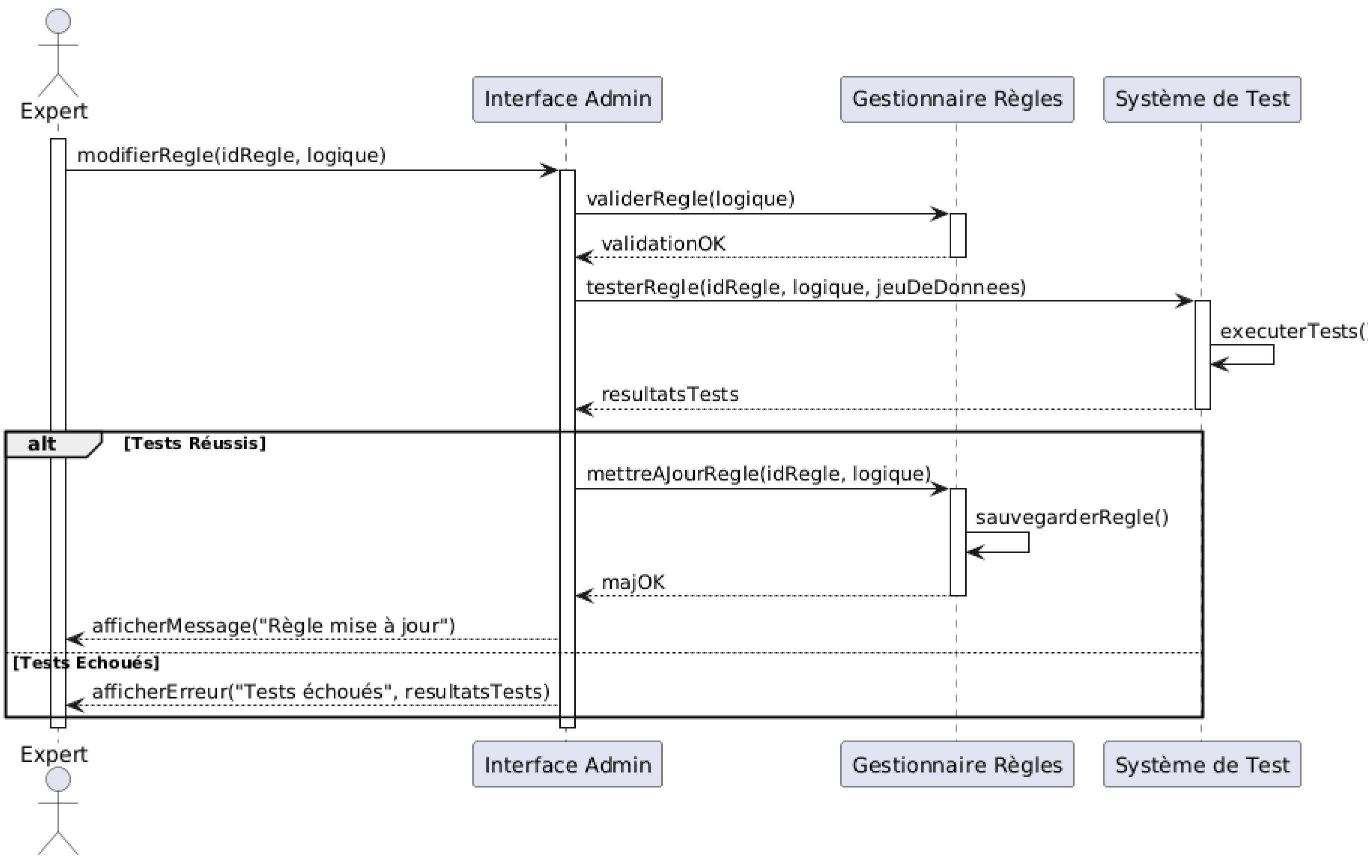

Fig. 4. Sequence diagram for scenario 2

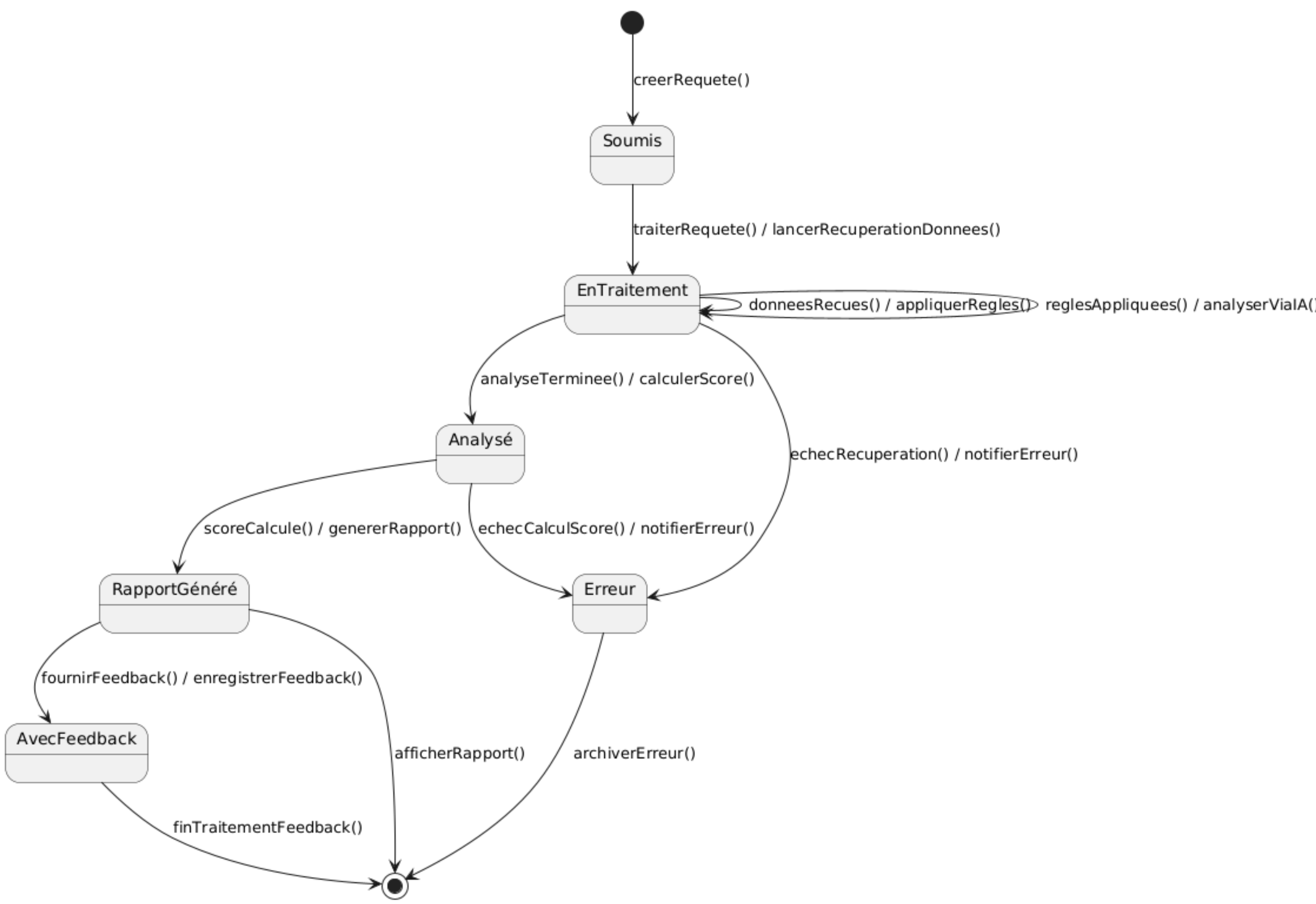

Fig. 5. State/transition diagram

- **Textual content-based rules:**
  - Detecting excessively emotional or sensationalist language (heavy use of capital letters, exclamation marks, superlatives) [23].
  - Checking for the presence (or blatant absence) of citations, references, or links to external sources to support claims.
  - Identifying typical structures of hoaxes or chain letters (e.g., "Share this massively...") [23].
  - Detecting certain simple logical fallacies if recurring linguistic patterns can be identified [28].
  - Basic stylometric analysis (sentence length, lexical diversity) compared to established norms for quality articles.
- **Metadata-based rules:**
  - Checking consistency between publication date and dates of reported events.
  - Flagging very old information presented as current without appropriate context.

The implementation of this engine could range from a series of simple conditional structures to the use of a dedicated rule engine (e.g., based on standards like Rete or using specific libraries). The role of the 'Expert' actor is fundamental here to define, refine, test, and weight these rules (attribute 'weight' in 'VerificationRule') to ensure their relevance and avoid false positives/negatives.

B. AI/NLP Module

The Artificial Intelligence and Natural Language Processing module ('AIModel' and 'NLPResult' in the UML model) is designed to perform deeper, semantic analyses of the content where simple rules show their limitations. It is essential for detecting more subtle forms of disinformation and for adapting to evolving tactics.

B.1. Text Representation

A fundamental preliminary step for any AI-based NLP analysis is the conversion of raw text into a numerical representation that models can process. Various techniques exist, ranging from traditional methods to modern approaches based on deep neural networks [2, 14, 19, 25, 29]:

- **Frequency-based methods:** TF-IDF (Term Frequency-Inverse Document Frequency) or Bag of Words (BoW) represent text based on word frequency, possibly weighted by their rarity in a corpus [14]. These methods are simple but lose word order and context.
- **Static Embeddings:** Word2Vec [25] or GloVe [2, 14, 29] learn dense vectors for each word, capturing some semantic relationships. However, a word has only one representation, regardless of its context.
- **Contextualized Embeddings:** Transformer-based models, like BERT [15, 36, 42] or Sentence-BERT (sBERT) [14], generate vector representations for words or sentences that depend on the context in which they appear [20]. These embeddings are generally more performant for tasks requiring fine-grained language understanding and have shown better stability across different datasets [14].

The choice of representation technique will depend on the specific AI models used for downstream tasks and the available computational resources.

B.2. Specific NLP Analyses

The AI module will integrate several types of analyses to assess different facets of credibility:

- **Sentiment Analysis:** Beyond simple positive/negative/neutral classification, the goal is to identify potentially manipulative use of emotions [2, 5, 11, 15, 37]. This includes detecting an excessively biased, inflammatory tone, or an emotional charge disproportionate to the subject matter. Contextual approaches, considering the domain (e.g., a neutral tone is expected in a scientific report, but not necessarily in an editorial), are preferable [2]. Models like bidirectional LSTMs or Transformers can be trained for this task [2, 15]. The results of this analysis contribute to assessing the impartiality and objectivity of the content [33].
- **Coherence Analysis:** Credible text is generally well-structured and thematically and logically coherent [4, 19, 25, 41]. Coherence analysis aims to detect breaks, contradictions, or lack of focus that could indicate low quality or fabrication. Different approaches can be combined:
  - *Local Coherence:* Analysis of transitions between adjacent sentences, for example by tracking entity flow (Entity Grid [4]) or using neural models trained to predict the relationship between sentences [4].
  - *Global Coherence:* Assessment of thematic consistency throughout the document. Techniques like topic modeling (e.g., LDA [2] or embedding-based approaches like BERTopic [41]) can be used to extract main themes and measure their distribution or evolution through the text (e.g., via JS divergence between thematic distributions of text segments [41]). Approaches based on sentence position prediction can also provide useful embeddings for assessing global structure [19, 25]. Coherence is related to perceived quality and professionalism of the source [33]. Low coherence has been observed as a potential indicator for fake news or AI-generated texts, although its standalone discriminatory power may be limited [41].
- **Bias Detection:** Biases (political, gender, racial, etc.) can undermine the objectivity and thus the credibility of information [3, 10, 17, 22, 36]. The system must attempt to identify these biases. Transformer models (BERT, RoBERTa...) can be fine-tuned on bias detection datasets [17, 22, 36]. Open-source tools and libraries also exist. Direct use of LLMs via prompting is another emerging approach [7, 17, 22, 36]. A major challenge is the availability of high-quality training data and managing class imbalance for less frequent bias types [36]. Bias detection is directly related to assessing impartiality [33].
- **Named Entity Recognition (NER):** NER identifies and categorizes key entities (people, organizations, locations, dates, etc.) in the text [37]. Its role in credibility assessment is multifaceted:
  - *Fact Verification:* Extracted entities can be used to query knowledge bases or search engines to verify factual claims about them.
  - *Authority Assessment:* Identifying cited experts or institutions helps assess whether the source relies on recognized authorities in the field [37]. The absence of such entities can be a negative signal.
  - *Analysis of Cited Sources:* If the article explicitly mentions its sources, NER can extract them for subsequent assessment of their own credibility [37].

– *Entity Bias Detection:* Analyzing the frequency and context of mentions of certain entities can reveal bias (e.g., disproportionate criticism or praise) [22, 36].

Table 2 summarizes the link between credibility indicators and the envisioned analysis techniques.

Integrating and weighting the results of these different analyses to arrive at a final credibility score (‘calculateScore‘ in ‘EvaluationSystem‘) is a major design challenge. An ensemble learning approach, where the outputs of different models (rules, sentiment, coherence, bias, etc.) are combined, potentially with adaptive weights, could be considered [2].

C. Integration of External Data

The system does not operate in a vacuum; it critically depends on access to external information and services via APIs (‘ExternalSystem‘ in the UML model).

- **Search Engines (Google/Bing API):** These APIs are essential for retrieving additional context on a topic, finding articles corroborating or contradicting a claim, or identifying the original source of information [39]. Using these APIs requires obtaining keys [12, 24], managing queries (e.g., using quotes for exact searches, specifying language or region [24]), parsing JSON responses [12, 24], and especially rigorous management of usage quotas and associated costs, as well as potential errors (network, authentication, quota exceeded - code 429) [12, 24]. Third-party services like SerpApi can also aggregate these results.

- **Fact-Checking Databases / APIs:** Access to pre-existing fact-check databases is crucial to avoid reinventing the wheel. The Schema.org ‘ClaimReview‘ standard is widely used by fact-checking organizations to publish their assessments in a structured manner [35]. The Google Fact Check Tool API allows querying this database [13, 35], although it may have limitations on the information returned [13, 35]. Projects like ClaimsKG or CimpleKG aim to aggregate this data into knowledge graphs [27, 38]. Direct API access to specific sources like Snopes or PolitiFact seems more limited, often restricted to partnerships or internal tools [30]. Integrating these APIs allows the system to quickly check if a claim has already been addressed by professionals.

- **Large Language Model (LLM) APIs:** APIs from OpenAI (GPT), Google (Gemini), or Anthropic (Claude) [6] can be used for various NLP sub-tasks within the AI module, such as generating summaries of external articles, reformulating queries, generating explanations for the credibility score, or even classification tasks in zero-shot or few-shot mode if fine-tuned models are unavailable or too costly to develop [6, 20]. As with search engines, managing keys, costs, and usage limits is paramount.

The reliability and performance of the overall system are therefore intrinsically linked to the reliability, availability, and performance of these external APIs. A resilient architecture must include fallback mechanisms, caching, and fine-grained management of errors and limitations of these third-party services [12, 24]. Furthermore, the credibility of the information provided by these external systems (e.g., the accuracy of search results or the reliability of fact-checking databases) must also be considered in the overall calculation.

4. DISCUSSION

The UML modeling and description of technological components provide a conceptual basis for the credibility assessment system. This section discusses the implications of these choices, the inherent challenges in the domain, the limitations of the current model, and prospects for evolution.

A. Modeling Synthesis

The presented UML model articulates a modular architecture designed for extensibility and adaptability. Key actors (User, Expert, External Systems) and their main interactions are clearly defined through use cases. The static structure (class diagram) highlights the separation of concerns between request management, assessment orchestration, rule application, AI analysis, and external data access. The use of abstraction for external systems facilitates the addition of new data sources. Dynamic modeling (sequence and state diagrams) illustrates typical workflows for information assessment and system updates by an expert, confirming the conceptual feasibility of interactions. The hybrid approach, combining rules and AI, is explicitly supported by the proposed architecture.

B. Strengths and Limitations of the Model

The proposed model has several strengths:

- **Modularity and Extensibility:** The clear separation of components (rules, AI, external access, report) and the use of principles like abstraction make the system potentially easier to maintain and extend with new features, rules, AI models, or data sources [9].

- **Explicit Hybrid Approach:** The model recognizes the need to combine logical and connectionist approaches, reflecting the state of the art and the limitations of purely AI-based or rule-based systems [2].

- **Adaptability via the Expert:** The inclusion of the Expert actor and associated use cases allows for continuous system adaptation, essential in the dynamic field of disinformation [26, 42].

- **Richness of Assessment:** The structure of ‘EvaluationReport‘ allows for capturing and potentially presenting a multidimensional view of credibility, going beyond a simple binary score.

- **Feedback Loop:** Considering user feedback (UC7, ‘WithFeedback‘ state) opens the way for continuous improvement based on real user experience.

However, the model also has limitations inherent at this design stage:

- **Abstraction of Internal Complexity:** The precise logic within the rule engine and AI models is not detailed. The complexity of semantic analysis, bias detection, or coherence modeling is largely abstracted [9].

- **Unspecified Hybrid Integration Mechanism:** Although the structure supports the hybrid approach, the exact way rule and AI results are combined to calculate the final score is not defined in the UML model. Is it a weighted sum? A voting system? Do rules filter AI input? This remains a crucial design decision.

- **Performance and Scalability:** The UML model does not directly address non-functional aspects such as response time, throughput, or the ability to handle a large volume of concurrent requests. The efficiency of external API calls and the complexity of AI models will have a major impact on these aspects.

- **User Interface:** The user interface (UI) and user experience (UX) are not modeled. How (potentially complex) credibility information is presented in a "simple and understandable" manner is a major design challenge in itself.

Table 2. Key Credibility Indicators and Associated Analysis Techniques

Credibility Indicator	Associated Analysis Techniques in the Proposed System
Accuracy / Factuality [30, 33]	- Verification via Fact-Checking API (e.g., Google Fact Check API / ClaimReview) [13, 35] - NER + Querying knowledge bases / Search engines [37] - Analysis of presence and quality of citations/references [28]
Authority / Source Reputation [30, 33]	- Consultation of source reputation databases (e.g., via rules based on MediaBias/-FactCheck) [30] - Analysis of source history (via rules, e.g., domain age, past publications) - NER to identify cited experts/institutions [37]
Objectivity / Impartiality / Bias [30, 33]	- Bias Detection Models (Political, Gender, etc.) [7, 17, 22, 36] - Contextualized Sentiment Analysis (detection of manipulative/excessive tone) [2] - Analysis of diversity of viewpoints presented (via retrieval of external info)
Presentation / Style / Quality [33]	- Coherence Analysis (Local and Global) [4, 41] - Rules for detecting sensationalist or unprofessional language [23] - Stylometric analysis (lexical diversity, syntactic complexity) [2] - Detection of AI-generated text (as a potential signal) [20, 28]
Timeliness / Currency [33]	- Rules for checking date consistency (publication vs. events) - Flagging outdated information presented as current
Persuasion Techniques / Fallacies [28, 30]	- Rules for detecting known patterns (e.g., excessive emotional appeals, simple fallacies) [23, 28] - Sentiment Analysis (to detect emotional manipulation) [2]

C. Research Challenges in Credibility Assessment

The development of such a system is part of an active research field facing fundamental challenges:

- **Volume, Velocity, Variety:** The very nature of the web and social media generates an incessant flow of multifaceted information (text, image, video) at a speed that defies analysis capabilities, even automated ones [14, 20]. Designing systems capable of processing this scale in near real-time is a major technical challenge.

- **The Evolving Impact of GenAI:** The growing ability of GenAI to produce high-quality synthetic content poses a dynamic threat [20, 42]. Disinformation techniques are constantly evolving, requiring continuous adaptation of detection systems. This creates a kind of technological "arms race" where defenders must constantly innovate to counter new attack methods.

- **Multimodality:** Disinformation often combines text, images, and videos to enhance its impact [20, 35]. The current model focuses primarily on text. A comprehensive credibility assessment will require multimodal analysis capabilities in the future, capable of detecting inconsistencies or manipulations across different media formats.

- **Context Dependence and Subjectivity:** Credibility is not an intrinsic and absolute property. It strongly depends on context (domain, target audience, communication intent) and involves a degree of subjectivity [30, 39]. For example, a claim may be technically true but presented misleadingly (framing bias). Managing this nuance, distinguishing opinion from factual assertion, and correctly interpreting intent (e.g., satire vs. disinformation) are considerable challenges for automated systems.

- **The Gap between Automation and Human Judgment:** Despite AI advancements, many tasks related to credibility assessment, particularly in specialized fields like medicine or for complex claims, still require nuanced human judgment [14, 26, 30]. Finding the right balance between automation for scalability and human intervention for accuracy and reliability remains an open challenge [34]. The proposed system integrates an Expert for configuration, but human involvement in the assessment process itself might be necessary for borderline cases.

D. Challenges Related to Data and Evaluation

Beyond conceptual challenges, very practical obstacles exist regarding data and evaluation methods in this field:

- **Quality of Training Data:** Recent studies have highlighted serious problems in many datasets used to train and evaluate disinformation detection models [38]. These problems include:
  - *Spurious Correlations:* Models may learn to predict veracity based on irrelevant signals, such as the presence of certain keywords (e.g., politicians' names) or the data collection period, rather than actual content [38]. This leads to models that perform well on the specific dataset but do not generalize to new data.
  - *Ambiguity and Infeasibility:* Many claims in datasets lack sufficient context to be verified, even with access to web search [38]. Training models on such data amounts to asking them to guess, which does not measure their actual ability to assess credibility.

  The development of the proposed system will need to pay particular attention to the curation and validation of data used for training and testing AI components.

- **Limitations of Standard Evaluation Metrics:** Classic metrics like accuracy or F1-score, based on comparing categorical labels (true/false), prove insufficient for evaluating modern systems, especially those based on GenAI [38]. A system can produce valid reasoning and a relevant explanation while assigning a final label that differs from the dataset's "ground truth" (e.g., if the ground truth is outdated). It is necessary to develop more nuanced metrics and evaluation protocols, which could include assessing the quality of generated explanations or using LLMs to evaluate reasoning consistency [38]. The Evaluation Quality Assessment (EQA) approach, proposing a critical analysis of the data and evaluation methods used, is a promising avenue [38].

- **Cost and Difficulty of Annotation:** Obtaining high-quality annotations for credibility, especially when requiring domain expertise (as in medicine), is a costly and time-consuming process [14, 30, 38]. Techniques like active annotation, which aim to optimize annotators' time by prioritizing the most informative or most likely non-credible examples, can help mitigate this problem [14, 26].

Table 3 illustrates the performance variability even for a specific task like bias detection, highlighting the challenges related to data and evaluation.

E. Ethical Considerations

The development and deployment of a credibility assessment system raise important ethical questions:

- **Algorithmic Bias:** The system's AI components can unintentionally learn and amplify biases present in training data (e.g., associating certain topics or demographic groups with lower credibility) [7, 22]. Regular audits and bias mitigation techniques are necessary to ensure fairness.

- **Transparency and Explainability:** To be genuinely useful and trustworthy, the system must be able to explain why it assigns a certain credibility score [2, 20]. Providing only a numerical score without justification risks being counterproductive. The design of 'EvaluationReport' must integrate clear explanatory elements.

- **Effect on Critical Thinking:** A potential risk is that users become overly reliant on the tool and stop exercising their own critical judgment [26]. The interface design and how results are presented should aim to *support* and *encourage* critical thinking, rather than replace it, for example by highlighting key indicators and uncertainties.

- **Potential for Misuse:** Like any powerful tool, a credibility assessment system could be misused, for example, to systematically target and discredit legitimate but critical sources, or to reinforce echo chambers by validating only information conforming to a certain viewpoint. Governance and oversight mechanisms are important.

F. Future Work

This modeling work is a first step. Subsequent steps necessary to realize the system include:

- **Detailed Development:** Implementing the complex internal logic of rules and AI models (training, fine-tuning). Developing the precise interaction and weighting mechanism for the hybrid approach.

- **Technical Implementation:** Coding the different classes and their interactions, managing external dependencies (APIs), ensuring robustness and performance.

- **User Interface:** Designing and developing an intuitive user interface that presents results effectively and promotes critical thinking.

- **Rigorous Evaluation:** Conducting thorough evaluations using high-quality datasets (or explicitly acknowledging the limitations of available datasets) and appropriate evaluation metrics, going beyond simple F1 scores [38]. User studies will also be necessary to assess the real impact on users' perception and judgment.

- **Multimodal Extension:** Integrating image and video analysis for more comprehensive coverage of modern disinformation.

- **Continuous Improvement:** Implementing processes for regular updates of rules and AI models by experts, in response to evolving threats and feedback.

- **Research on Explainability:** Developing methods to generate clear and reliable explanations of the credibility assessments produced by the system.

In conclusion, the presented UML modeling provides a coherent structure for a complex and relevant system. However, its effective realization will require overcoming significant technical, research, and ethical challenges, particularly concerning data quality, robust evaluation, and human-machine interaction that fosters critical thinking.

REFERENCES

1. Adali, S., Escriva, R., Goldberg, M. K., Magdon-Ismail, M., Szymanski, B. K., Wallace, W. A., and Williams, G. (2015). Measuring behavioral trust in social networks. In *2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 153–160.
2. Ahmed, T., Traore, I., and Saad, S. (2024). AI-Driven Hybrid Model for Fake News Detection: Integrating NLP, Sentiment Analysis, and Source Credibility to Combat Misinformation. *ResearchGate (Preprint)*. Accessed April 2025, DOI might be available later.
3. Baly, R., Da San Martino, G., Glass, J., and Nakov, P. (2020). We can detect your bias: Predicting the political ideology of news articles. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6785–6791.
4. Barzilay, R. and Lapata, M. (2008). Modeling local coherence: An entity-based approach. *Computational Linguistics*, 34(1):1–34.
5. Castillo, C., Mendoza, M., and Poblete, B. (2011). Information credibility on twitter. In *Proceedings of the 20th International Conference on World Wide Web (WWW '11)*, pages 675–684.
6. Chen, K. and Shu, K. (2023). Combating misinformation in the age of llms: Opportunities and challenges.
7. Chen, W.-F. and Oh, A. (2020). Detecting media bias using gaussian mixture models: An unsupervised approach.
8. de Souza, D. F., Castro, L. H. M., Oliveira, D., and Zarpelão, B. B. (2020). A survey on the automation of fake news detection. *Information Fusion*, 62:1–26.
9. Fowler, M. (2003). *UML Distilled: A Brief Guide to the Standard Object Modeling Language*. Addison-Wesley, Boston, MA, USA, 3rd edition.
10. Garg, S., Perot, V., Limtiaco, N., Taly, A., D'Amour, A., and Hooker, S. (2019). Counterfactual fairness in text classification through robustness. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 219–226.
11. Giachanou, A. and Crestani, F. (2016). Like it or not: A survey of twitter sentiment analysis methods. *ACM Computing Surveys (CSUR)*, 49(2):1–41.

Table 3. Illustrative Example of Transformer Model Performance for Multiple Bias Detection (Based on [36])

Bias Type	Metric	BERT	RoBERTa	ALBERT	DistilBERT	XLNet
Political	F1-Score	**0.89**	0.87	0.85	0.84	0.86
Gender	F1-Score	**0.82**	0.80	0.75	0.73	0.76
Entity	F1-Score	**0.85**	0.84	0.81	0.80	0.82
Racial	F1-Score	**0.65**	0.62	0.55	0.38	0.51
Religious	F1-Score	**0.78**	0.77	0.72	0.70	0.74
Regional	F1-Score	**0.70**	0.68	0.63	0.59	0.65
Sensationalism	F1-Score	**0.80**	0.79	0.74	0.71	0.75
Average F1 Score (Macro)		**0.78**	0.77	0.72	0.68	0.73

Note: These values are illustrative and based on trends reported in [36]. Actual performance heavily depends on the specific dataset, fine-tuning, and class imbalance management. Lower scores for Racial and Regional biases reflect challenges posed by data imbalance mentioned in the study.

12. Google. Custom search json api. Google Developers Documentation. Accessed April 2025.
13. Google. Fact check tools api. Google Developers Documentation. Accessed April 2025.
14. Hagag, B. and Tsarfaty, R. (2023). The truth, the whole truth, and nothing but the truth: A new benchmark dataset for hebrew text credibility assessment. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3850–3865, Singapore. Association for Computational Linguistics.
15. He, L., Hu, S., and Pei, A. (2023). Debunking disinformation: Revolutionizing truth with nlp in fake news detection.
16. Kedzie, C., McKeown, K., and Diaz, F. (2018). Content-driven detection of false news. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING 2018)*, pages 1771–1783, Santa Fe, New Mexico, USA. Association for Computational Linguistics.
17. Krieger, F. M., Ziegele, M., Jackisch, J., and Jakobs, I. (2021). Daroberta: Domain-adaptive roberta for detecting media bias. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2675–2681.
18. Larman, C. (2004). *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development*. Prentice Hall, Upper Saddle River, NJ, USA, 2nd edition.
19. Li, J. and Hovy, E. (2014). A model of coherence based on distributed sentence representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2039–2048.
20. Loth, A., Kappes, M., and Pahl, M.-O. (2024). Blessing or curse? a survey on the impact of generative ai on fake news. v2, 27 Dec 2024.
21. Lounis, H. (2025). Modélisation des problèmes complexes en sciences cognitives. Présentations et diapositives du cours DIC-9251, UQAM.
22. Menzner, P. and Leidner, J. L. (2024). Biasscanner: Detecting biased statements in news articles.
23. Metzger, M. J., Flanagin, A. J., and Medders, R. B. (2010). Social and heuristic approaches to credibility evaluation online. *Journal of Communication*, 60(3):413–439.
24. Microsoft (2025). Bing search apis documentation. https://learn.microsoft.com/en-us/azure/cognitive-services/bing-web-search. Accessed April 2025.
25. Mikolov, T., Chen, K., Corrado, G. S., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
26. Nabozny, A., Marcinkiewicz, M., Nielek, R., and Wierzbicki, A. (2021). Active annotation in evaluating the credibility of web-based medical information: Guidelines for creating training data sets for machine learning. *JMIR Medical Informatics*, 9(11):e26065.
27. Osborne, F., Recupero, D. R., Buscaldi, D., and Motta, E. (2024). Cimplekg: A knowledge graph for linking claims with explanations and context. ISWC 2024 Resources Track Submission.
28. Oshikawa, R., Qian, J., and Wang, W. Y. (2020). A survey on natural language processing for fake news detection. *arXiv preprint arXiv:1811.00770*. v3.
29. Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.
30. Pescuma, V. N., Osborne, F., Recupero, D. R., and Motta, E. (2025). Source credibility assessment: A comprehensive survey. In *International Journal of Interactive Multimedia and Artificial Intelligence*.
31. Prabhakaran, V., Hutchinson, B., and Mitchell, M. (2019). Perturbation sensitivity analysis to detect unintended model biases.
32. Premaratne, P., Hotaling, J., Tignor, S., Blasch, E., Kadar, I., Kokar, M., Matheus, C., Salerno, J., and Stytz, M. (2012). Consensus algorithms for credibility assessment of soft information. In *Proceedings of the 4th International Conference on Applied Human Factors and Ergonomics (AHFE)*, San Francisco, CA, USA.
33. Rieh, S. Y. (2010). Credibility assessment of online information in context. *Information Research*, 15(3). paper 445.
34. Sattar, F., Shabbir, M., Ahmad, M. A., Ghazanfar, M. A., and Axelsson, J. (2020). Claimeval: Integrated framework for joint source credibility estimation and claim evaluation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 1024–1031.
35. Schema.org Community (2025). Claimreview - schema.org type. https://schema.org/ClaimReview. Accessed April 2025.
36. Shah, B. S., Shah, D. S., and Attar, V. (2025). Decoding news bias: Multi bias detection in news articles. v1, 5 Jan 2025.
37. Sharma, K., Qian, F., Jiang, H., Ruchansky, N., Zhang, M., and Liu, Y. (2019). Combating fake news: A survey on identification and mitigation techniques. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(3):1–42.
38. Thibault, C., Tian, J.-J., Peloquin-Skulski, G., Curtis, T. L., Zhou, J., Laflamme, F., Guan, Y., Rabbany, R., Godbout, J.-F., and Pelrine, K. (2025). A guide to misinformation detection data and evaluation. v2, 19 Mar 2025.
39. Viviani, M. and Pasi, G. (2017). Credibility in social media: opinions, news, and health information—a survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 7(5):e1209.
40. W3C Credibility Community Group (2018). Credibility signals (draft community group report). W3C Community Group Draft Report.
41. Willems, R. (2025). Modeling thematic coherence: An interpretable approach for analyzing fake and llm-generated news. Master's thesis, Utrecht University, Utrecht, The Netherlands. MSc AI Thesis.
42. Zhou, X. and Zafarani, R. (2020). A survey of fake news: Fundamen-

tal theories, detection methods, and opportunities. *ACM Computing Surveys (CSUR)*, 53(5):1–40.

CONTENTS

A. APPENDIX 0: THE INTERFACE

B. APPENDIX I (PYTHON CODE OF MY HYBRID SYSTEM - PROTOTYPE)

Listing 1. Prototype Python Code for Hybrid System

```
import re
import requests # Gard  pour d' ventuels  appels API r els
from transformers import pipeline, AutoTokenizer, AutoModelForSequenceClassification
import numpy as np
import torch # N cessaire pour certains mod les transformers

# # #############################
# # code in Python by
# # Dominique S. Loyer
# # May 2025
# # #############################
# # Please use the citation key
# #  if you use the code
# # Citation Key: loyerModelingHybridSystem2025
# #############################
# #############################
# #############################
# LIME est conserv  pour l'explicabilit , mais d'autres techniques pourraient  tre  n cessaires
# pour diff rents types de mod les (ex: SHAP).
from lime.lime_text import LimeTextExplainer
from urllib.parse import urlparse # Pour analyser les URLs
import datetime # Pour la date de g n ration du rapport

# --- Configuration Initiale (Mod les et Explainers) ---
# On charge les mod les ici pour  viter  de les recharger    chaque appel.
# NOTE : Pour une application r elle, envisagez des mod les plus sp cifiques
#         pour la d tection de biais, la coh rence, etc.
#         Certains mod les peuvent n cessiter un fine-tuning.

# Mod le de sentiment (comme dans votre code original)
sentiment_pipeline = pipeline("sentiment-analysis", model="distilbert-base-uncased-finetuned-sst-2-english")

# Mod le pour la d tection de biais (Exemple - n cessite un mod le appropri )
# Remplacer par un mod le  entra n  pour la d tection de biais.
# Exemple : 'd4data/bias-detection-model' (v rifier disponibilit  sur Hugging Face Hub)
# Pour l'instant, on utilise un mod le de classification g n rique comme placeholder.
bias_tokenizer = AutoTokenizer.from_pretrained("bert-base-uncased")
bias_model = AutoModelForSequenceClassification.from_pretrained("bert-base-uncased") # PLACEHOLDER

# Mod le pour la Reconnaissance d'Entit s Nomm es (NER)
ner_pipeline = pipeline("ner",              ="dbmdz/bert-large-cased-finetuned-conll03-english", grouped_entities=True) # grouped_entities est souvent utile, added specific model for better results

# Explainer LIME (pour le mod le de sentiment pour l'instant)
# Note : L'explicabilit  pour d'autres mod les (ex: biais) n cessiterait une configuration adapt e.
# The original line "explainer = LimeTextExplainer(class_names=)" is incomplete Python.
# Assuming sentiment model has 'NEGATIVE', 'POSITIVE' classes. Adjust if different.
explainer = LimeTextExplainer(class_names=['NEGATIVE', 'POSITIVE'])

# --- Fonctions Utilitaires ---

def is_url(text):
    """V rifie si une cha ne ressemble    une URL."""
    try:
        result = urlparse(text)
        return all([result.scheme, result.netloc])
    except ValueError:
        return False

def fetch_web_content(url):
    """
    Simule la r cup ration du contenu textuel d'une URL.
    Pour une impl mentation r elle, utiliser `requests` et `BeautifulSoup`.
    """
    print(f"␣R cup ration␣du␣contenu␣de␣:␣{url}")
    # Simuler diff rents contenus pour tester
    if "verified-news.com" in url:
        return "This␣official␣report␣is␣verified␣and␣credible.␣All␣facts␣checked."
    elif "hoax-site.org" in url:
        return "Shocking␣conspiracy␣revealed!␣Experts␣are␣wrong.␣This␣is␣a␣hoax!"
    else:
        # Simuler le cas o   une URL ne retourne rien ou est inaccessible
        if "nonexistent-domain-for-test.xyz" in url:
            print(f"␣ chec ␣de␣la␣r cup ration␣pour␣:␣{url}")
            return None # Simule un  chec
        return "Some␣generic␣content␣from␣the␣web."

def fetch_external_data(text_or_url):
    """
    Simule la r cup ration de donn es externes (fact-checking, r putation source).
    Pour une impl mentation r elle, appeler des API (Google Fact Check, NewsGuard, etc.).
    """
    print(f"␣Recherche␣de␣donn es␣externes␣pour␣:␣{str(text_or_url)[:50]}...") # Assurer que c'est une str pour le slicing
    external_info = {
        'fact_checks': [],
        'source_reputation': 'Unknown',
        'domain_age_days': None, # Initialis     None
        'related_articles': []
    }
    # Tente de r cup rer les infos uniquement si c'est une URL valide
    if isinstance(text_or_url, str) and is_url(text_or_url):
```

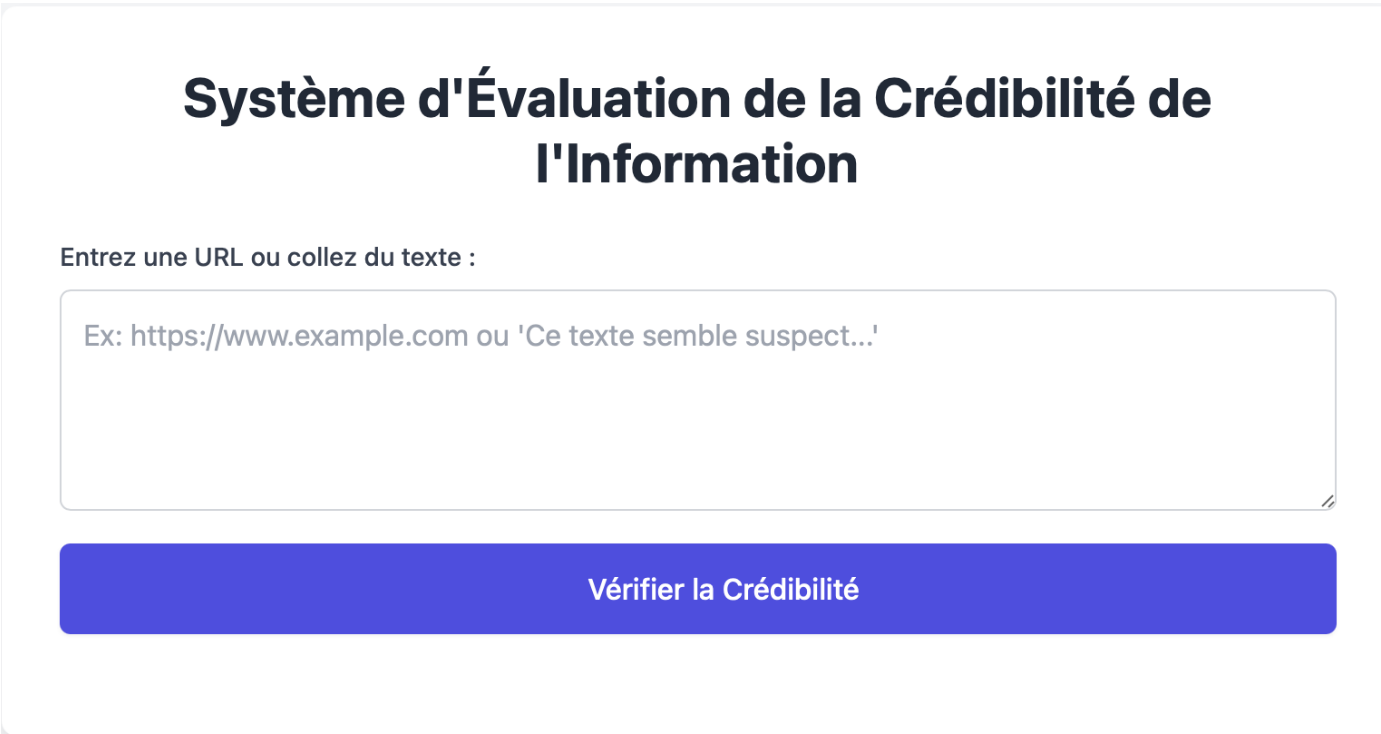

Fig. 6. The simplistic Interface (in french only as in May 2025

```python
        domain = urlparse(text_or_url).netloc
        if "verified-news.com" in domain:
            external_info['source_reputation'] = 'High'
            external_info['domain_age_days'] = 1500 # D fini seulement pour les URLs reconnues
            external_info['fact_checks'].append({'claim': 'Official␣report␣facts', 'rating': 'True'})
        elif "hoax-site.org" in domain:
            external_info['source_reputation'] = 'Low'
            external_info['domain_age_days'] = 90 # D fini seulement pour les URLs reconnues
            external_info['fact_checks'].append({'claim': 'Conspiracy␣theory', 'rating': 'False'})
        elif "nonexistent-domain-for-test.xyz" not in domain: # Ne pas donner d' ge pour le domaine inexistant
            external_info['source_reputation'] = 'Medium'
            external_info['domain_age_days'] = 730 # D fini seulement pour les URLs reconnues

    # Simulation de r sultats de recherche (peut  tre  ajout  m me si ce n'est pas une URL)
    external_info['related_articles'] = ["Article␣A␣on␣similar␣topic.", "Article␣B␣with␣different␣view."]
    return external_info

# --- Classe Principale du Syst me ---

class CredibilityVerificationSystem:
    def __init__(self):
        # Les mod les sont charg s globalement, on peut les r f rencer ici si besoin
        self.sentiment_pipeline = sentiment_pipeline
        self.ner_pipeline = ner_pipeline
        self.bias_tokenizer = bias_tokenizer
        self.bias_model = bias_model
        self.explainer = explainer

    def preprocess(self, text):
        """Nettoyage simple du texte."""
        # Am liorable : suppression de HTML, normalisation unicode, etc.
        if not isinstance(text, str): # V rifier si l'entr e est bien une cha ne
            return ""
        text = re.sub(r'http\S+|www\S+|https\S+', '', text, flags=re.MULTILINE) # Enlever les URLs
        text = re.sub(r'\s+', '␣', text) # Normaliser les espaces
        text = re.sub(r'[^\w\s\.\?,!]', '', text) # Garder ponctuation basique
        return text.lower().strip()

    def rule_based_analysis(self, text, external_data):
        """
        Analyse bas e sur des r gles logiques pr d finies et des donn es externes.
        Ceci est une version simplifi e bas e sur le PDF.
        """
        results = {
            'linguistic_markers': {},
            'source_analysis': {},
            'timeliness_flags': []
        }
        # 1. Marqueurs Linguistiques (Exemples simples)
        sensational_words = ['shocking', 'revealed', 'conspiracy', 'amazing', 'secret']
        certainty_words = ['verified', 'authentic', 'credible', 'proven', 'fact']
        doubt_words = ['hoax', 'false', 'fake', 'unproven', 'rumor']

        results['linguistic_markers']['sensationalism'] = sum(1 for word in sensational_words if word in text)
        results['linguistic_markers']['certainty'] = sum(1 for word in certainty_words if word in text)
        results['linguistic_markers']['doubt'] = sum(1 for word in doubt_words if word in text)

        # 2. Analyse de la Source (bas e sur les donn es externes simul es)
        results['source_analysis']['reputation'] = external_data.get('source_reputation', 'Unknown')
        domain_age = external_data.get('domain_age_days') # R cup rer la valeur (peut  tre  None)
        results['source_analysis']['domain_age_days'] = domain_age # Stocker la valeur r cup r e

        # 3. Actualit  (Exemple tr s basique)
        # *** CORRECTION ICI ***
        # V rifier si domain_age n'est PAS None AVANT de comparer
        if domain_age is not None and domain_age < 180: # Moins de 6 mois
            results['timeliness_flags'].append('Source␣domain␣is␣relatively␣new.')

        # 4. V rification des Faits (Fact-Checking)
        results['fact_checking'] = external_data.get('fact_checks', [])

        return results

    def nlp_analysis(self, text):
        """
        Analyse via des mod les NLP (IA).
        """
        results = {
            'sentiment': None,
            'sentiment_explanation': None,
            'bias_analysis': {'score': None, 'label': 'Unavailable'}, # Placeholder
            'named_entities': None,
            'coherence_score': None # Placeholder
        }

        # V rification suppl mentaire si le texte est vide apr s preprocess
        if not text:
            print("Avertissement␣:␣Texte␣vide␣fourni␣  ␣nlp_analysis.")
            results['sentiment'] = {'label': 'Neutral', 'score': 0.5} # Ou une autre valeur par d faut
            return results # Retourner les r sultats par d faut

        # 1. Analyse de Sentiment (avec explicabilit  LIME)
        try:
            # Pr diction pour LIME
            def predict_proba_sentiment(texts):
                # S'assurer que texts est une liste de cha nes
                if isinstance(texts, str):
                    texts = [texts]
                elif not isinstance(texts, list):
                    texts = list(texts) # Tenter de convertir en liste

                processed_texts = [self.preprocess(t) for t in texts]
```

```
                # G rer les textes vides apr s pr traitement
                valid_texts = [t for t in processed_texts if t]
                probabilities = []

                if not valid_texts:
                    # Retourner une distribution neutre pour chaque texte original si tous sont vides
                    return np.array([[0.5, 0.5]] * len(texts))

                # Faire la pr diction uniquement sur les textes valides
                # The pipeline returns a list of dicts, e.g., [{'label': 'POSITIVE', 'score': 0.99}]
                predictions = self.sentiment_pipeline(valid_texts)
                pred_idx = 0
                for original_text_processed in processed_texts: # Iterate based on original number of texts
                    if original_text_processed: # If the text was not empty after preprocessing
                        pred = predictions[pred_idx]
                        # Assurer que la sortie est toujours [prob_neg, prob_pos]
                        # For distilbert-base-uncased-finetuned-sst-2-english, 'POSITIVE' and 'NEGATIVE'
                        if pred['label'] == 'POSITIVE':
                            probabilities.append([1 - pred['score'], pred['score']])
                        else: # NEGATIVE (or map other labels if model is different)
                            probabilities.append([pred['score'], 1 - pred['score']])
                        pred_idx += 1
                    else:
                        probabilities.append([0.5, 0.5]) # Probabilit  neutre pour texte vide
                return np.array(probabilities)

            # Obtenir la pr diction principale pour le texte unique
            main_prediction_list = self.sentiment_pipeline(text) # pipeline returns a list
            results['sentiment'] = main_prediction_list[0] if main_prediction_list else {'label': 'Error', 'score': 0.0}


            # G n rer l'explication LIME
            explanation = self.explainer.explain_instance(
                text,
                predict_proba_sentiment,
                num_features=6 # Nombre de mots/features   montrer dans l'explication
            )
            results['sentiment_explanation'] = explanation.as_list()

        except Exception as e:
            print(f"Erreur lors de l'analyse de sentiment ou LIME : {e}")
            results['sentiment'] = {'label': 'Error', 'score': 0.0}
            results['sentiment_explanation'] = []

        # 2. Analyse de Biais (Simulation/Placeholder)
        # Un vrai mod le de d tection de biais serait n cessaire ici.
        try:
            inputs = self.bias_tokenizer(text, return_tensors="pt", truncation=True, max_length=512, padding=True)
            with torch.no_grad():
                logits = self.bias_model(**inputs).logits
            # Assuming the first logit corresponds to "no bias" and second to "bias" for a 2-class model
            # This is a placeholder, actual model output needs to be interpreted correctly
            simulated_bias_score = torch.softmax(logits, dim=1)[0][1].item() # Prob of bias class
            if simulated_bias_score > 0.7: # Seuil arbitraire
                results['bias_analysis'] = {'score': simulated_bias_score, 'label': 'Potential Bias Flagged (Simulated)'}
            else:
                results['bias_analysis'] = {'score': simulated_bias_score, 'label': 'Low Bias Detected (Simulated)'}
        except Exception as e:
            print(f"Erreur lors de l'analyse de biais (simul e) : {e}")
            results['bias_analysis'] = {'score': None, 'label': 'Error'}


        # 3. Reconnaissance d'Entit s Nomm es (NER)
        try:
            entities = self.ner_pipeline(text)
            results['named_entities'] = entities
        except Exception as e:
            print(f"Erreur lors de l'analyse NER : {e}")
            results['named_entities'] = []

        # 4. Analyse de Coh rence (Placeholder)
        results['coherence_score'] = np.random.rand() # Score al atoire pour l'exemple

        return results

    def calculate_overall_score(self, rule_results, nlp_results):
        """
        Calcule un score de cr dibilit  global bas  sur les analyses.
        Ceci est une heuristique simple,   affiner consid rablement.
        Le score va de 0 (peu cr dible)   1 (tr s cr dible).
        """
        score = 0.5 # Score de base neutre

        # Max possible positive adjustment and negative adjustment
        # This is a simplistic way to bound the score between 0 and 1
        # A more robust approach would use weighted sums and normalization
        positive_score_factors = 0
        negative_score_factors = 0

        # --- Pond rations (Exemples -   AJUSTER ABSOLUMENT) ---
        WEIGHT_REPUTATION_HIGH = 0.15
        WEIGHT_REPUTATION_LOW = -0.20
        WEIGHT_AGE_OLD = 0.05
        WEIGHT_AGE_NEW = -0.10
        WEIGHT_CERTAINTY_HIGH = 0.05 # If many certainty words and no doubt words
        WEIGHT_DOUBT_HIGH = -0.15   # If doubt words are present
        WEIGHT_SENSATIONALISM_HIGH = -0.10
        WEIGHT_SENTIMENT_VERY_NEG = -0.05
        WEIGHT_BIAS_FLAGGED = -0.15
        WEIGHT_COHERENCE_HIGH = 0.03 # Low weight as it's simulated
        WEIGHT_COHERENCE_LOW = -0.03

        # Facteurs bas s sur les r gles
        source_rep = rule_results['source_analysis']['reputation']
        if source_rep == 'High':
            positive_score_factors += WEIGHT_REPUTATION_HIGH
        elif source_rep == 'Low':
            negative_score_factors += WEIGHT_REPUTATION_LOW # Note: this is a negative value

        domain_age = rule_results['source_analysis'].get('domain_age_days')
        if domain_age is not None:
            if domain_age > 365 * 2: # Ex: > 2 ans
                positive_score_factors += WEIGHT_AGE_OLD
            elif domain_age < 90: # Ex: < 3 mois
                negative_score_factors += WEIGHT_AGE_NEW

        certainty_count = rule_results['linguistic_markers']['certainty']
        doubt_count = rule_results['linguistic_markers']['doubt']
        if certainty_count > 1 and doubt_count == 0:
            positive_score_factors += WEIGHT_CERTAINTY_HIGH
        if doubt_count > 0:
            negative_score_factors += WEIGHT_DOUBT_HIGH * min(doubt_count, 2) # Cap penalty

        sensationalism_count = rule_results['linguistic_markers']['sensationalism']
        if sensationalism_count > 0:
            negative_score_factors += WEIGHT_SENSATIONALISM_HIGH * min(sensationalism_count, 2) # Cap penalty

        # Facteurs bas s sur le NLP
        sentiment_info = nlp_results.get('sentiment')
        if sentiment_info and sentiment_info.get('label') == 'NEGATIVE' and sentiment_info.get('score', 0) > 0.85:
            negative_score_factors += WEIGHT_SENTIMENT_VERY_NEG

        bias_info = nlp_results.get('bias_analysis')
        if bias_info and 'Flagged' in bias_info.get('label', '') and bias_info.get('score', 0) > 0.7:
             negative_score_factors += WEIGHT_BIAS_FLAGGED * (bias_info['score'] - 0.5) * 2 # Scale impact

        coherence_val = nlp_results.get('coherence_score') # Simulated 0 to 1
        if coherence_val is not None:
            if coherence_val > 0.7:
                positive_score_factors += WEIGHT_COHERENCE_HIGH
            elif coherence_val < 0.3:
                negative_score_factors += WEIGHT_COHERENCE_LOW

        # Combine factors: start from 0.5 and add/subtract scaled factors
        # This is still heuristic. A proper model would learn weights.
        final_score = 0.5 + positive_score_factors + negative_score_factors

        # Clamp score between 0 and 1
        final_score = max(0.0, min(1.0, final_score))

        return final_score

    def generate_report(self, input_info, rule_results, nlp_results, overall_score):
        """G n re un rapport structur ."""
        report = {
            "timestamp": datetime.datetime.now().isoformat(),
            "input_information": str(input_info)[:200] + "..." if len(str(input_info)) > 200 else str(input_info), # Truncate long inputs
            "overall_credibility_score": f"{overall_score:.2f}",
            "summary": "", # Sera rempli ci-dessous
            "details": {
                "rule_based_analysis": rule_results,
                "nlp_analysis": nlp_results,
                "external_data_summary": { # R sum  des donn es externes
                    "source_reputation": rule_results.get('source_analysis',{}).get('reputation','N/A'),
                    "domain_age_days": rule_results.get('source_analysis',{}).get('domain_age_days','N/A'),
                    "fact_checks_found": len(rule_results.get('fact_checking',[]))
                }
            },
            "explanation_notes": []
        }

        # G n rer un r sum  simple
        if overall_score >= 0.75:
            report["summary"] = "The information appears to be generally credible based on the analysis."
            report["explanation_notes"].append("High credibility score suggests reliability.")
        elif overall_score >= 0.5:
            report["summary"] = "The information has mixed indicators of credibility. Caution is advised."
            report["explanation_notes"].append("Moderate score. Review details for specific concerns.")
        elif overall_score >= 0.25:
            report["summary"] = "The information shows several indicators of low credibility. Approach with significant caution."
            report["explanation_notes"].append("Low score. Multiple red flags identified.")
        else:
            report["summary"] = "The information appears to have very low credibility based on the analysis."
            report["explanation_notes"].append("Very low score. Likely unreliable.")

        # Ajouter des notes sp cifiques bas es sur les r sultats
        if rule_results['source_analysis']['reputation'] == 'Low':
            report["explanation_notes"].append("Source reputation is assessed as low.")
        if rule_results['linguistic_markers']['sensationalism'] > 1:
```

```
            report["explanation_notes"].append("High use of sensationalist language detected.")
        if nlp_results.get('bias_analysis', {}).get('label', '').startswith('Potential_Bias'):
            report["explanation_notes"].append("Potential bias flagged in the content.")
        if not nlp_results.get('named_entities'):
             report["explanation_notes"].append("Few or no specific named entities (people, organizations, locations) found, which can sometimes indicate vague or unsubstantiated claims.")

        # Simplifier l'explication LIME pour le rapport
        if nlp_results.get('sentiment_explanation'):
            simplified_lime = []
            for word, weight in nlp_results['sentiment_explanation']:
                simplified_lime.append(f"'{word}' (impact: {weight:.2f})")
            nlp_results['sentiment_explanation'] = ", ".join(simplified_lime)


        return report

    def verify_credibility(self, text_or_url):
        """Orchestre le processus de v rification."""
        print(f"\n--- V rification de : {str(text_or_url)[:100]} ---") # Assurer str pour slicing

        content_to_analyze = text_or_url
        input_is_url = is_url(text_or_url)

        if input_is_url:
            print("Input est une URL. Tentative de r cup ration du contenu web.")
            web_content = fetch_web_content(text_or_url)
            if web_content:
                content_to_analyze = web_content
            else:
                print("Impossible de r cup rer le contenu web, analyse de l'URL elle-m me et des m tadonn es.")
                # Si le contenu web n'est pas r cup rable, on analyse l'URL elle-m me (moins d'infos)
                # ou on pourrait retourner une erreur/un score bas directement.
                # Pour l'instant, on continue avec l'URL comme "texte".
                content_to_analyze = text_or_url # Fallback to analyzing the URL string if content fetch fails

        # R cup rer les donn es externes (bas  sur l'URL originale si c'en est une, sinon sur le texte)
        external_data = fetch_external_data(text_or_url if input_is_url else content_to_analyze)

        # Pr traitement du contenu textuel
        processed_text = self.preprocess(content_to_analyze)

        if not processed_text and input_is_url and not web_content:
            print("Avertissement : Aucun contenu textuel   analyser pour l'URL et le contenu n'a pas pu  tre r cup r .")
            # Attribuer un score bas si aucune info textuelle ne peut  tre analys e
            # Et que les donn es externes sont aussi minimales.
            # Ceci est une heuristique, une meilleure gestion d'erreur serait n cessaire.
            dummy_rules = self.rule_based_analysis("", external_data) # Analyse de r gles avec texte vide
            dummy_nlp = {'sentiment': {'label': 'Neutral', 'score': 0.5}, 'bias_analysis': {'label': 'Unavailable'}, 'named_entities': [], 'coherence_score': 0.5, 'sentiment_explanation':[]}
            low_score = self.calculate_overall_score(dummy_rules, dummy_nlp)
            # Ajuster le score encore plus bas si la r putation de la source est inconnue ou basse
            if external_data.get('source_reputation', 'Unknown') in ['Unknown', 'Low']:
                 low_score = min(low_score, 0.1) # Force un score tr s bas
            return self.generate_report(text_or_url, dummy_rules, dummy_nlp, low_score)


        # Analyse bas e sur les r gles
        rule_results = self.rule_based_analysis(processed_text, external_data)
        print(f"R sultats R gles: {rule_results}")

        # Analyse NLP (IA)
        # S'assurer que processed_text n'est pas vide avant de le passer   NLP
        if not processed_text:
             print("Avertissement: Texte vide apr s pr traitement, NLP limit .")
             # Fournir des r sultats NLP par d faut si le texte est vide
             nlp_results = {'sentiment': {'label': 'Neutral', 'score': 0.5}, 'bias_analysis': {'label': 'Unavailable'}, 'named_entities': [], 'coherence_score': 0.5, 'sentiment_explanation':[]}
        else:
            nlp_results = self.nlp_analysis(processed_text)
        print(f"R sultats NLP: {nlp_results}")


        # Calcul du score global
        overall_score = self.calculate_overall_score(rule_results, nlp_results)
        print(f"Score Global Calcul : {overall_score:.2f}")

        # G n ration du rapport
        report = self.generate_report(text_or_url, rule_results, nlp_results, overall_score)

        return report

# --- Exemple d'Utilisation ---
if __name__ == "__main__":
    system = CredibilityVerificationSystem()

    test_inputs = [
        "This is an amazing article full of verified facts from scientists.",
        "URGENT! Shocking conspiracy revealed by an insider! Everything you know is a lie!",
        "http://verified-news.com/article123",
        "http://hoax-site.org/exclusive-story",
        "http://example.com/some-neutral-content",
        "http://nonexistent-domain-for-test.xyz/page.html", # URL qui  chouera   r cup rer du contenu
        "Just a short neutral statement.",
        "", # Test avec une cha ne vide
        "According to a new study published in Nature, climate change is accelerating. The study, involving 200 scientists, points to rising sea levels. However, some critics argue the models are flawed. It is important to consider multiple perspectives. The Prime Minister commented today."
    ]

    for inp in test_inputs:
        final_report = system.verify_credibility(inp)
        print("\n--- RAPPORT FINAL ---")
        for key, value in final_report.items():
            if isinstance(value, dict):
                print(f"{key}:")
                for sub_key, sub_value in value.items():
                    print(f"  {sub_key}: {sub_value}")
            elif isinstance(value, list) and key == "explanation_notes":
                 print(f"{key}:")
                 for item in value:
                     print(f"  - {item}")
            else:
                print(f"{key}: {value}")
        print("--------------------\n")

    # Test sp cifique pour LIME avec un texte qui pourrait avoir des mots positifs/n gatifs clairs
    lime_test_text = "This product is absolutely fantastic and wonderful, but the customer service was terrible and awful."
    print(f"\n--- Test LIME sp cifique pour: '{lime_test_text}' ---")
    processed_lime_text = system.preprocess(lime_test_text)
    if processed_lime_text:
        nlp_res_lime = system.nlp_analysis(processed_lime_text)
        print("Sentiment:", nlp_res_lime.get('sentiment'))
        print("Sentiment Explanation (LIME):", nlp_res_lime.get('sentiment_explanation'))
    else:
        print("Texte vide apr s pr traitement pour le test LIME.")
```

C. APPENDIX II: PREDICATE LOGIC RULES (CONCEPTUAL)

This appendix outlines conceptual examples of predicate logic rules that was part of the rule-based engine. [6]

A. Source Reputation Rules

- $\forall x, \mathrm{IsURL}(x) \land \mathrm{DomainInBlacklist}(\mathrm{GetDomain}(x)) \implies \mathrm{CredibilityModifier}(x, \mathrm{negative}, \mathrm{strong})$
- $\forall x, \mathrm{IsURL}(x) \land \mathrm{DomainInWhitelist}(\mathrm{GetDomain}(x)) \implies \mathrm{CredibilityModifier}(x, \mathrm{positive}, \mathrm{moderate})$
- $\forall x, \mathrm{IsURL}(x) \land \mathrm{HasContactInfo}(\mathrm{GetContent}(x)) \land \mathrm{HasEditorialPolicy}(\mathrm{GetContent}(x)) \implies \mathrm{CredibilityModifier}(x, \mathrm{positive}, \mathrm{slight})$
- $\forall x, \mathrm{IsURL}(x) \land \mathrm{DomainAge}(\mathrm{GetDomain}(x), \mathrm{days}) \land \mathrm{days} < 90 \implies \mathrm{Flag}(x, \mathrm{RecentDomain})$

B. Content-Based Rules

- $\forall x, \mathrm{IsText}(x) \land \mathrm{ContainsExcessiveCaps}(\mathrm{GetContent}(x)) \implies \mathrm{Flag}(x, \mathrm{SensationalLanguage})$
- $\forall x, \mathrm{IsText}(x) \land \mathrm{CountSensationalKeywords}(\mathrm{GetContent}(x), N) \land N > 3 \implies \mathrm{CredibilityModifier}(x, \mathrm{negative}, \mathrm{moderate})$
- $\forall x, \mathrm{IsText}(x) \land \mathrm{LacksCitations}(\mathrm{GetContent}(x)) \land \mathrm{IsClaimType}(\mathrm{GetContent}(x), \mathrm{Factual}) \implies \mathrm{Flag}(x, \mathrm{UnsupportedClaims})$
- $\forall x, \mathrm{IsText}(x) \land \mathrm{ContainsPattern}(x, \text{"Share this immediately!"}) \implies \mathrm{Flag}(x, \mathrm{HoaxPattern})$

C. Fact-Checking Integration Rules

- $\forall x, \mathrm{IsClaim}(x) \land \mathrm{ExistsFactCheck}(x, \mathrm{rating}, \mathrm{sourceFC}) \land \mathrm{rating} = \text{"False"} \implies \mathrm{CredibilityModifier}(x, \mathrm{negative}, \mathrm{very\_strong})$

[6] see Appendice III

- $\forall x, \text{IsURL}(x) \wedge \text{AssociatedClaimReview}(x, \text{review}) \wedge \text{GetRating}(review) = \text{"False"} \implies \text{CredibilityModifier}(x, \text{negative}, \text{very\_strong})$

Note: As a completetly fonctional system would need more nuanced predicates, ways to handle uncertainty, and a mechanism to combine evidence from multiple rules. The ‘GetContent(x)‘, ‘GetDomain(x)‘, ‘DomainInBlacklist‘ etc. would be functions querying the system's data and knowledge. [7]

D. APPENDICE III TURTLE CODE TO POPULATE THE ONTOLOGY(SUBJECT-PREDICATE-OBJECT)

```
[language=Turtle,basicstyle=\ttfamily\tiny,frame=single]

@base <http://www.dic9335.uqam.ca/ontologies/credibility-verification#> .
@prefix : <http://www.dic9335.uqam.ca/ontologies/credibility-verification#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
#
# # #############################
# # code in Turtle by
# # Dominique S. Loyer
# # May 2025
# # #############################
# # Please use the citation key
# #  if you use the code
# # Citation Key: loyerModelingHybridSystem2025
# #############################
# #############################
# #############################
# #
# #    Annotation properties
# #
# #############################
#
#
# http://www.w3.org/2002/07/owl#maxCardinality
#
#
#
# #######################
# #
# #    Object Properties
# #
# ########################
#
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#analyzesSource
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#appliesRule
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#assignsCredibilityLevel
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#basedOnEvidence
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#concernsCriterion
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#concernsInformation
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#configuredByExpert
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#evaluatesCriterion
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#fetchesDataFrom
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#hasAuthor
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#hasCriterionResult
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#hasOriginalSource
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#includesNLPResult
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#includesRuleResult
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#includesSourceAnalysis
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#isReportOf
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#isSubjectOfRequest
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#obtainedVia
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#originatesFrom
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#producesReport
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#submitsRequest
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#submittedBy
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#usesModel
#
#
#
# ######################
# #    Data properties
# #
# #####################
#
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#authorName
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#coherenceScore
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#completionTimestamp
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#credibilityLevelValue
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#credibilityScoreValue
```

[7] See Appendice III for the Turtle code that was impleted thus far

```
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#criterionResultConfidence
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#criterionResultValue
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#detectedBiases
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#evidenceSnippet
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#evidenceURL
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#informationContent
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#informationURL
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#modelName
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#modelType
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#reportSummary
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#requestStatus
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#ruleDescription
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#ruleLogic
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#ruleResultValid
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#ruleWeight
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#sentimentScore
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#sourceAnalyzedReputation
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#sourceAnalyzedURL
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#sourceMentionsCount
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#sourceReputationScore
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#sourceURL
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#submissionTimestamp
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#userName
#
#
#
# ########################
# #    Classes
# #
# #######################
#
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#AcademicJournal
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#ApiLLM
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#Author
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#BaseDeFaits
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#CredibilityLevel
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#Evidence
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#Expert
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#FactCheckingOrganization
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#InfoSourceAnalyse
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#InformationFaibleCredibilite
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#InformationHauteCredibilite
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#InformationMoyenneCredibilite
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#InformationSoumise
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#InformationVerifiee
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#ModeleIA
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#MoteurRecherche
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#NewsWebsite
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#Niveau_Bas
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#Niveau_Haut
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#Niveau_Moyen
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#Niveau_NonVerifie
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#PersonalBlog
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#RapportEvaluation
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#RefutingEvidence
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#RegleVerification
```

```
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#RequeteEvaluation
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#ResultatCritere
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#ResultatNLP
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#ResultatRegle
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#ResultatVerification
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#SocialMediaPlatform
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#Source
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#SupportingEvidence
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#SystemeExterne
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#User
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#VerificationCriterion
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#VerificationMethod
#
#
#
# ######################
# #
# #    Individuals
# #
# #####################
#
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#Criteria_AuthorExpertise
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#Criteria_CoherenceAnalysis
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#Criteria_CrossReferencing
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#Criteria_FactCheckDB
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#Criteria_SourceReputation
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#Criteria_ToneAnalysis
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#Niveau_Bas
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#Niveau_Haut
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#Niveau_Moyen
#
# http://www.dic9335.uqam.ca/ontologies/credibility-verification#Niveau_NonVerifie
#
#
#
# ##########################
# #
# #    Annotations
# #
# ##########################
#
#
#
#
#
#
#
#
# ########################
# #
# #    General axioms
# #
# ######################
#
#
#
#
#
#

<credibility-verification> a owl:Ontology;
  rdfs:comment "Ontologie␣enrichie␣et␣adapt e␣mod lisant␣les␣concepts␣li s␣ ␣la␣v rification␣de␣la␣cr dibilit ␣des␣sources␣d'information␣sur␣le␣Web,␣bas e␣sur␣le␣rapport␣de␣mod lisation␣UML␣et␣inspir e␣par␣l'ontologie␣de␣subvention␣recherche."@fr;
  rdfs:label "Ontologie␣Syst me␣de␣V rification␣de␣Sources␣(Adapt e␣Rapport␣+␣Subvention)"@fr;
  owl:versionInfo "2.1" .

owl:maxCardinality a owl:AnnotationProperty .

:analyzesSource a owl:ObjectProperty;
  rdfs:domain :InfoSourceAnalyse;
  rdfs:range :Source;
  rdfs:label "analyse␣source"@fr .

:appliesRule a owl:ObjectProperty, owl:FunctionalProperty;
  rdfs:domain :ResultatRegle;
  rdfs:range :RegleVerification;
  rdfs:label "applique␣r gle"@fr .

:assignsCredibilityLevel a owl:ObjectProperty, owl:FunctionalProperty;
  rdfs:domain :RapportEvaluation;
  rdfs:range :CredibilityLevel;
  rdfs:comment "Lie␣un␣rapport␣d' valuation ␣au␣niveau␣de␣cr dibilit ␣final␣attribu ."@fr;
  rdfs:label "assigne␣niveau␣cr dibilit "@fr .

:basedOnEvidence a owl:ObjectProperty;
  rdfs:domain :RapportEvaluation;
  rdfs:range :Evidence;
  rdfs:comment "Lie␣un␣rapport␣d' valuation ␣aux␣preuves␣collect es."@fr;
  rdfs:label "bas ␣sur␣preuve"@fr .

:concernsCriterion a owl:ObjectProperty, owl:FunctionalProperty;
  rdfs:domain :ResultatCritere;
  rdfs:range :VerificationCriterion;
  rdfs:label "concerne␣crit re"@fr .

:concernsInformation a owl:ObjectProperty, owl:FunctionalProperty;
  owl:inverseOf :isSubjectOfRequest;
  rdfs:domain :RequeteEvaluation;
```

```
  rdfs:range :InformationSoumise;
  rdfs:label "concerne␣information"@fr .

:configuredByExpert a owl:ObjectProperty;
  rdfs:domain _:genid1;
  rdfs:range :Expert;
  rdfs:label "configur ␣par␣expert"@fr .

_:genid1 a owl:Class;
  owl:unionOf _:genid4 .

_:genid4 a rdf:List;
  rdf:first :ModeleIA;
  rdf:rest _:genid3 .

_:genid3 a rdf:List;
  rdf:first :RegleVerification;
  rdf:rest _:genid2 .

_:genid2 a rdf:List;
  rdf:first :VerificationCriterion;
  rdf:rest rdf:nil .

:evaluatesCriterion a owl:ObjectProperty;
  rdfs:domain _:genid5;
  rdfs:range :VerificationCriterion;
  rdfs:comment "Lie␣une␣r gle␣ou␣un␣mod le␣au␣
      crit re␣de␣v rification␣qu'il␣est␣con u␣
      pour␣ valuer ."@fr;
  rdfs:label " value ␣crit re"@fr .

_:genid5 a owl:Class;
  owl:unionOf _:genid7 .

_:genid7 a rdf:List;
  rdf:first :ModeleIA;
  rdf:rest _:genid6 .

_:genid6 a rdf:List;
  rdf:first :RegleVerification;
  rdf:rest rdf:nil .

:fetchesDataFrom a owl:ObjectProperty;
  rdfs:domain :RequeteEvaluation;
  rdfs:range :SystemeExterne;
  rdfs:label "r cup re␣donn es␣de"@fr .

:hasAuthor a owl:ObjectProperty;
  rdfs:domain :InformationSoumise;
  rdfs:range :Author;
  rdfs:comment "Lie␣une␣information␣soumise␣ ␣son␣
      auteur␣pr sum ."@fr;
  rdfs:label "a␣pour␣auteur"@fr .

:hasCriterionResult a owl:ObjectProperty;
  rdfs:domain :RapportEvaluation;
  rdfs:range :ResultatCritere;
  rdfs:comment "Lie␣un␣rapport␣au␣r sultat␣
      d taill ␣pour␣un␣crit re␣d' valuation ␣
      sp cifique."@fr;
  rdfs:label "a␣r sultat␣pour␣crit re"@fr .

:hasOriginalSource a owl:ObjectProperty;
  rdfs:domain :InformationSoumise;
  rdfs:range :Source;
  rdfs:comment "Lie␣une␣information␣soumise␣ ␣sa␣
      source␣d'origine␣principale."@fr;
  rdfs:label "a␣pour␣source␣originale"@fr .

:includesNLPResult a owl:ObjectProperty;
  rdfs:domain :RapportEvaluation;
  rdfs:range :ResultatNLP;
  rdfs:label "inclut␣r sultat␣NLP"@fr .

:includesRuleResult a owl:ObjectProperty;
  rdfs:domain :RapportEvaluation;
  rdfs:range :ResultatRegle;
  rdfs:label "inclut␣r sultat␣r gle"@fr .

:includesSourceAnalysis a owl:ObjectProperty;
  rdfs:domain :RapportEvaluation;
  rdfs:range :InfoSourceAnalyse;
  rdfs:label "inclut␣analyse␣source"@fr .

:isReportOf a owl:ObjectProperty, owl:
    InverseFunctionalProperty;
  owl:inverseOf :producesReport;
  rdfs:domain :RapportEvaluation;
  rdfs:range :RequeteEvaluation;
  rdfs:label "est␣rapport␣de"@fr .

:isSubjectOfRequest a owl:ObjectProperty;
  rdfs:domain :InformationSoumise;
  rdfs:range :RequeteEvaluation;
  rdfs:label "est␣sujet␣de␣requ te"@fr .

:obtainedVia a owl:ObjectProperty;
  rdfs:domain :ResultatCritere;
  rdfs:range _:genid8;
  rdfs:label "obtenu␣via"@fr .

_:genid8 a owl:Class;
  owl:unionOf _:genid10 .

_:genid10 a rdf:List;
  rdf:first :ResultatNLP;
  rdf:rest _:genid9 .

_:genid9 a rdf:List;
  rdf:first :ResultatRegle;
  rdf:rest rdf:nil .

:originatesFrom a owl:ObjectProperty;
  rdfs:domain :Evidence;
  rdfs:range :Source;
  rdfs:comment "Lie␣une␣preuve␣ ␣la␣source␣d'o ␣
      elle␣a␣ t ␣extraite."@fr;
  rdfs:label "provient␣de"@fr .

:producesReport a owl:ObjectProperty, owl:
    FunctionalProperty;
  rdfs:domain :RequeteEvaluation;
  rdfs:range :RapportEvaluation;
  rdfs:label "produit␣rapport"@fr .

:submitsRequest a owl:ObjectProperty;
  owl:inverseOf :submittedBy;
  rdfs:domain :User;
  rdfs:range :RequeteEvaluation;
  rdfs:label "soumet␣requ te"@fr .

:submittedBy a owl:ObjectProperty, owl:
    FunctionalProperty;
  rdfs:domain :RequeteEvaluation;
  rdfs:range :User;
  rdfs:comment "Lie␣une␣requ te␣de␣v rification␣
      ␣l'utilisateur␣qui␣l'a␣soumise."@fr;
  rdfs:label "soumise␣par"@fr .

:usesModel a owl:ObjectProperty, owl:
    FunctionalProperty;
  rdfs:domain :ResultatNLP;
  rdfs:range :ModeleIA;
  rdfs:label "utilise␣mod le"@fr .

:authorName a owl:DatatypeProperty;
  rdfs:domain :Author;
  rdfs:range xsd:string;
  rdfs:label "nom␣de␣l'auteur"@fr .

:coherenceScore a owl:DatatypeProperty;
  rdfs:domain :ResultatNLP;
```

```
  rdfs:range xsd:float;
  rdfs:label "score␣coh rence"@fr .

:completionTimestamp a owl:DatatypeProperty, owl:
    FunctionalProperty;
  rdfs:domain :RapportEvaluation;
  rdfs:range xsd:dateTime;
  rdfs:label "horodatage␣de␣compl tion"@fr .

:credibilityLevelValue a owl:DatatypeProperty, owl:
    FunctionalProperty;
  rdfs:domain :CredibilityLevel;
  rdfs:range xsd:float;
  rdfs:label "valeur␣num rique␣niveau"@fr .

:credibilityScoreValue a owl:DatatypeProperty, owl:
    FunctionalProperty;
  rdfs:domain :RapportEvaluation;
  rdfs:range xsd:float;
  rdfs:label "valeur␣score␣cr dibilit "@fr .

:criterionResultConfidence a owl:DatatypeProperty;
  rdfs:domain :ResultatCritere;
  rdfs:range xsd:float;
  rdfs:label "confiance␣r sultat␣crit re"@fr .

:criterionResultValue a owl:DatatypeProperty;
  rdfs:domain :ResultatCritere;
  rdfs:range xsd:string;
  rdfs:label "valeur␣r sultat␣crit re"@fr .

:detectedBiases a owl:DatatypeProperty;
  rdfs:domain :ResultatNLP;
  rdfs:range xsd:string;
  rdfs:comment "";
  rdfs:label "biais␣d tect s"@fr .

:evidenceSnippet a owl:DatatypeProperty;
  rdfs:domain :Evidence;
  rdfs:range xsd:string;
  rdfs:label "extrait␣de␣la␣preuve"@fr .

:evidenceURL a owl:DatatypeProperty;
  rdfs:domain :Evidence;
  rdfs:range xsd:anyURI;
  rdfs:label "URL␣de␣la␣preuve"@fr .

:informationContent a owl:DatatypeProperty;
  rdfs:domain :InformationSoumise;
  rdfs:range xsd:string;
  rdfs:label "contenu␣de␣l'information"@fr .

:informationURL a owl:DatatypeProperty;
  rdfs:domain :InformationSoumise;
  rdfs:range xsd:anyURI;
  rdfs:label "URL␣de␣l'information"@fr .

:modelName a owl:DatatypeProperty;
  rdfs:domain :ModeleIA;
  rdfs:range xsd:string;
  rdfs:label "nom␣mod le"@fr .

:modelType a owl:DatatypeProperty;
  rdfs:domain :ModeleIA;
  rdfs:range xsd:string;
  rdfs:label "type␣mod le"@fr .

:reportSummary a owl:DatatypeProperty;
  rdfs:domain :RapportEvaluation;
  rdfs:range xsd:string;
  rdfs:label "r sum ␣du␣rapport"@fr .

:requestStatus a owl:DatatypeProperty, owl:
    FunctionalProperty;
  rdfs:domain :RequeteEvaluation;
  rdfs:range xsd:string;
  rdfs:label "statut␣requ te"@fr .

:ruleDescription a owl:DatatypeProperty;
  rdfs:domain :RegleVerification;
  rdfs:range xsd:string;
  rdfs:label "description␣r gle"@fr .

:ruleLogic a owl:DatatypeProperty;
  rdfs:domain :RegleVerification;
  rdfs:range xsd:string;
  rdfs:label "logique␣r gle"@fr .

:ruleResultValid a owl:DatatypeProperty;
  rdfs:domain :ResultatRegle;
  rdfs:range xsd:boolean;
  rdfs:label "r sultat␣r gle␣valide"@fr .

:ruleWeight a owl:DatatypeProperty;
  rdfs:domain :RegleVerification;
  rdfs:range xsd:float;
  rdfs:label "poids␣r gle"@fr .

:sentimentScore a owl:DatatypeProperty;
  rdfs:domain :ResultatNLP;
  rdfs:range xsd:float;
  rdfs:label "score␣sentiment"@fr .

:sourceAnalyzedReputation a owl:DatatypeProperty;
  rdfs:domain :InfoSourceAnalyse;
  rdfs:range xsd:string;
  rdfs:label "r putation␣source␣analys e"@fr .

:sourceAnalyzedURL a owl:DatatypeProperty;
  rdfs:domain :InfoSourceAnalyse;
  rdfs:range xsd:anyURI;
  rdfs:label "URL␣source␣analys e"@fr .

:sourceMentionsCount a owl:DatatypeProperty;
  rdfs:domain :InfoSourceAnalyse;
  rdfs:range xsd:integer;
  rdfs:label "mentions␣source␣analys e"@fr .

:sourceReputationScore a owl:DatatypeProperty;
  rdfs:domain :Source;
  rdfs:range xsd:float;
  rdfs:label "score␣de␣r putation␣de␣la␣source"@fr
      .

:sourceURL a owl:DatatypeProperty, owl:
    FunctionalProperty;
  rdfs:domain :Source;
  rdfs:range xsd:anyURI;
  rdfs:label "URL␣de␣la␣source"@fr .

:submissionTimestamp a owl:DatatypeProperty, owl:
    FunctionalProperty;
  rdfs:domain :RequeteEvaluation;
  rdfs:range xsd:dateTime;
  rdfs:label "horodatage␣de␣soumission"@fr .

:userName a owl:DatatypeProperty;
  rdfs:domain :User;
  rdfs:range xsd:string;
  rdfs:label "nom␣d'utilisateur"@fr .

:AcademicJournal a owl:Class;
  rdfs:subClassOf :Source;
  rdfs:label "Revue␣Acad mique"@fr .

:ApiLLM a owl:Class;
  rdfs:subClassOf :SystemeExterne;
  rdfs:label "API␣de␣LLM"@fr .

:Author a owl:Class;
  rdfs:comment "Repr sente␣la␣personne␣ou␣l'entit
      ␣cr dit e␣pour␣la␣cr ation␣de␣l'
```

```
    information␣soumise."@fr;
  rdfs:label "Auteur"@fr .

:BaseDeFaits a owl:Class;
  rdfs:subClassOf :SystemeExterne;
  rdfs:label "Base␣de␣Donn es␣de␣Faits␣V rifi s"
    @fr .

:CredibilityLevel a owl:Class;
  rdfs:comment "Repr sente␣le␣niveau␣de␣
    cr dibilit ␣qualitatif␣ou␣quantitatif␣
    attribu ␣dans␣le␣rapport."@fr;
  rdfs:label "Niveau␣de␣Cr dibilit "@fr .

:Evidence a owl:Class;
  rdfs:comment "Repr sente␣un␣ lment ␣d'
    information␣externe␣utilis ␣pour␣ tayer ␣ou␣
    r futer␣l'information␣v rifi e."@fr;
  rdfs:label "Preuve"@fr .

:Expert a owl:Class;
  rdfs:subClassOf :User;
  rdfs:comment "Utilisateur␣qualifi ␣responsable␣de
    ␣la␣configuration␣et␣de␣l'am lioration␣du␣
    syst me␣(r gles,␣mod les)."@fr;
  rdfs:label "Expert"@fr .

:FactCheckingOrganization a owl:Class;
  rdfs:subClassOf :Source;
  rdfs:label "Organisation␣de␣V rification␣des␣
    Faits"@fr .

:InfoSourceAnalyse a owl:Class;
  rdfs:subClassOf _:genid11;
  rdfs:comment "D tails␣sur␣une␣source␣sp cifique␣
    telle␣qu'analys e␣et␣pr sent e␣dans␣le␣
    rapport."@fr;
  rdfs:label "Information␣Source␣Analys e"@fr .

_:genid11 a owl:Restriction;
  owl:cardinality "1"^^xsd:nonNegativeInteger;
  owl:onProperty :analyzesSource .

:InformationFaibleCredibilite a owl:Class;
  owl:equivalentClass _:genid12;
  rdfs:subClassOf _:genid22;
  rdfs:label "Information␣Faiblement␣Cr dible"@fr .

_:genid12 a owl:Class;
  owl:intersectionOf _:genid21 .

_:genid21 a rdf:List;
  rdf:first :InformationVerifiee;
  rdf:rest _:genid19 .

_:genid19 a rdf:List;
  rdf:first _:genid20;
  rdf:rest _:genid17 .

_:genid17 a rdf:List;
  rdf:first _:genid18;
  rdf:rest _:genid13 .

_:genid13 a rdf:List;
  rdf:first _:genid14;
  rdf:rest rdf:nil .

_:genid14 a owl:Restriction;
  owl:someValuesFrom _:genid15;
  owl:onProperty :isSubjectOfRequest .

_:genid15 a owl:Restriction;
  owl:someValuesFrom _:genid16;
  owl:onProperty :producesReport .

_:genid16 a owl:Restriction;
  owl:hasValue :Niveau_Bas;
  owl:onProperty :assignsCredibilityLevel .

_:genid18 a owl:Class;
  owl:complementOf :InformationMoyenneCredibilite .

_:genid20 a owl:Class;
  owl:complementOf :InformationHauteCredibilite .

_:genid22 a owl:Restriction;
  owl:allValuesFrom _:genid23;
  owl:onProperty :isSubjectOfRequest .

_:genid23 a owl:Restriction;
  owl:allValuesFrom _:genid24;
  owl:onProperty :producesReport .

_:genid24 a owl:Restriction;
  owl:hasValue :Niveau_Bas;
  owl:onProperty :assignsCredibilityLevel .

:InformationHauteCredibilite a owl:Class;
  owl:equivalentClass _:genid25;
  rdfs:subClassOf _:genid31;
  rdfs:label "Information␣Hautement␣Cr dible"@fr .

_:genid25 a owl:Class;
  owl:intersectionOf _:genid30 .

_:genid30 a rdf:List;
  rdf:first :InformationVerifiee;
  rdf:rest _:genid26 .

_:genid26 a rdf:List;
  rdf:first _:genid27;
  rdf:rest rdf:nil .

_:genid27 a owl:Restriction;
  owl:someValuesFrom _:genid28;
  owl:onProperty :isSubjectOfRequest .

_:genid28 a owl:Restriction;
  owl:someValuesFrom _:genid29;
  owl:onProperty :producesReport .

_:genid29 a owl:Restriction;
  owl:hasValue :Niveau_Haut;
  owl:onProperty :assignsCredibilityLevel .

_:genid31 a owl:Restriction;
  owl:allValuesFrom _:genid32;
  owl:onProperty :isSubjectOfRequest .

_:genid32 a owl:Restriction;
  owl:allValuesFrom _:genid33;
  owl:onProperty :producesReport .

_:genid33 a owl:Restriction;
  owl:hasValue :Niveau_Haut;
  owl:onProperty :assignsCredibilityLevel .

:InformationMoyenneCredibilite a owl:Class;
  owl:equivalentClass _:genid34;
  rdfs:subClassOf _:genid42;
  rdfs:label "Information␣Moyennement␣Cr dible"@fr
    .

_:genid34 a owl:Class;
  owl:intersectionOf _:genid41 .

_:genid41 a rdf:List;
  rdf:first :InformationVerifiee;
  rdf:rest _:genid39 .

_:genid39 a rdf:List;
  rdf:first _:genid40;
```

```
  rdf:rest _:genid35 .

_:genid35 a rdf:List;
  rdf:first _:genid36;
  rdf:rest rdf:nil .

_:genid36 a owl:Restriction;
  owl:someValuesFrom _:genid37;
  owl:onProperty :isSubjectOfRequest .

_:genid37 a owl:Restriction;
  owl:someValuesFrom _:genid38;
  owl:onProperty :producesReport .

_:genid38 a owl:Restriction;
  owl:hasValue :Niveau_Moyen;
  owl:onProperty :assignsCredibilityLevel .

_:genid40 a owl:Class;
  owl:complementOf :InformationHauteCredibilite .

_:genid42 a owl:Restriction;
  owl:allValuesFrom _:genid43;
  owl:onProperty :isSubjectOfRequest .

_:genid43 a owl:Restriction;
  owl:allValuesFrom _:genid44;
  owl:onProperty :producesReport .

_:genid44 a owl:Restriction;
  owl:hasValue :Niveau_Moyen;
  owl:onProperty :assignsCredibilityLevel .

:InformationSoumise a owl:Class;
  rdfs:comment "Repr sente␣l'unit ␣d'information␣(
      texte,␣URL)␣telle␣que␣soumise␣pour␣
      v rification."@fr;
  rdfs:label "Information␣Soumise"@fr .

:InformationVerifiee a owl:Class;
  owl:equivalentClass _:genid45;
  rdfs:label "Information␣V rifi e"@fr .

_:genid45 a owl:Class;
  owl:intersectionOf _:genid49 .

_:genid49 a rdf:List;
  rdf:first :InformationSoumise;
  rdf:rest _:genid46 .

_:genid46 a rdf:List;
  rdf:first _:genid47;
  rdf:rest rdf:nil .

_:genid47 a owl:Restriction;
  owl:someValuesFrom _:genid48;
  owl:onProperty :isSubjectOfRequest .

_:genid48 a owl:Restriction;
  owl:someValuesFrom :RapportEvaluation;
  owl:onProperty :producesReport .

:ModeleIA a owl:Class;
  rdfs:subClassOf :VerificationMethod, _:genid50;
  rdfs:comment "Repr sente␣un␣mod le␣d'
      apprentissage␣automatique␣utilis ␣pour␣l'
      analyse␣s mantique␣ou␣autre."@fr;
  rdfs:label "Mod le␣IA/NLP"@fr .

_:genid50 a owl:Restriction;
  owl:minCardinality "1"^^xsd:nonNegativeInteger;
  owl:onProperty :evaluatesCriterion .

:MoteurRecherche a owl:Class;
  rdfs:subClassOf :SystemeExterne;
  rdfs:label "Moteur␣de␣Recherche"@fr .

:NewsWebsite a owl:Class;
  rdfs:subClassOf :Source;
  rdfs:label "Site␣d'actualit s"@fr .

:Niveau_Bas a owl:Class, owl:NamedIndividual, :
    CredibilityLevel;
  :credibilityLevelValue "0.2"^^xsd:float;
  rdfs:label "Cr dibilit ␣Faible"@fr .

:Niveau_Haut a owl:Class, owl:NamedIndividual, :
    CredibilityLevel;
  :credibilityLevelValue "0.8"^^xsd:float;
  rdfs:label "Cr dibilit ␣ leve "@fr .

:Niveau_Moyen a owl:Class, owl:NamedIndividual, :
    CredibilityLevel;
  :credibilityLevelValue "0.5"^^xsd:float;
  rdfs:label "Cr dibilit ␣Moyenne"@fr .

:Niveau_NonVerifie a owl:Class, owl:NamedIndividual,
     :CredibilityLevel;
  rdfs:label "Non␣V rifi "@fr .

:PersonalBlog a owl:Class;
  rdfs:subClassOf :Source;
  rdfs:label "Blog␣Personnel"@fr .

:RapportEvaluation a owl:Class;
  rdfs:subClassOf _:genid51;
  rdfs:comment "Encapsule␣les␣r sultats␣complets␣du
      ␣processus␣de␣v rification␣pour␣une␣requ te
      ␣donn e."@fr;
  rdfs:label "Rapport␣d' valuation "@fr .

_:genid51 a owl:Restriction;
  owl:cardinality "1"^^xsd:nonNegativeInteger;
  owl:onProperty :assignsCredibilityLevel .

:RefutingEvidence a owl:Class;
  rdfs:subClassOf :Evidence;
  owl:disjointWith :SupportingEvidence;
  rdfs:label "Preuve␣r futante"@fr .

:RegleVerification a owl:Class;
  rdfs:subClassOf :VerificationMethod, _:genid52;
  rdfs:comment "Repr sente␣une␣r gle␣logique␣
      pr d finie␣utilis e␣pour␣ valuer ␣un␣
      aspect␣de␣la␣cr dibilit ."@fr;
  rdfs:label "R gle␣de␣V rification"@fr .

_:genid52 a owl:Restriction;
  owl:minCardinality "1"^^xsd:nonNegativeInteger;
  owl:onProperty :evaluatesCriterion .

:RequeteEvaluation a owl:Class;
  rdfs:subClassOf _:genid53, _:genid54, _:genid55;
  rdfs:comment "Repr sente␣une␣demande␣sp cifique␣
      de␣v rification␣de␣cr dibilit ␣soumise␣par
      ␣un␣utilisateur."@fr;
  rdfs:label "Requ te␣d' valuation "@fr .

_:genid53 a owl:Restriction;
  owl:minCardinality "0"^^xsd:nonNegativeInteger;
  owl:onProperty :producesReport .

_:genid54 a owl:Restriction;
  owl:cardinality "1"^^xsd:nonNegativeInteger;
  owl:onProperty :concernsInformation .

_:genid55 a owl:Restriction;
  owl:cardinality "1"^^xsd:nonNegativeInteger;
  owl:onProperty :submittedBy .

:ResultatCritere a owl:Class;
  rdfs:subClassOf _:genid56, _:genid57;
```

```
  rdfs:comment "Repr sente␣le␣r sultat␣de␣l'
       valuation ␣d'un␣crit re␣sp cifique␣pour␣
      une␣requ te,␣potentiellement␣bas ␣sur␣un␣ou
      ␣plusieurs␣r sultats␣de␣r gles/NLP."@fr;
  rdfs:label "R sultat␣Crit re"@fr .

_:genid56 a owl:Restriction;
  owl:minCardinality "1"^^xsd:nonNegativeInteger;
  owl:onProperty :obtainedVia .

_:genid57 a owl:Restriction;
  owl:cardinality "1"^^xsd:nonNegativeInteger;
  owl:onProperty :concernsCriterion .

:ResultatNLP a owl:Class;
  rdfs:subClassOf :ResultatVerification, _:genid58;
  owl:disjointWith :ResultatRegle;
  rdfs:comment "R sultat␣de␣l'analyse␣effectu e␣
      par␣un␣mod le␣IA/NLP."@fr;
  rdfs:label "R sultat␣NLP"@fr .

_:genid58 a owl:Restriction;
  owl:cardinality "1"^^xsd:nonNegativeInteger;
  owl:onProperty :usesModel .

:ResultatRegle a owl:Class;
  rdfs:subClassOf :ResultatVerification, _:genid59;
  rdfs:comment "R sultat␣de␣l'application␣d'une␣
      r gle␣de␣v rification␣sp cifique."@fr;
  rdfs:label "R sultat␣R gle"@fr .

_:genid59 a owl:Restriction;
  owl:cardinality "1"^^xsd:nonNegativeInteger;
  owl:onProperty :appliesRule .

:ResultatVerification a owl:Class;
  rdfs:comment "Classe␣parente␣pour␣les␣r sultats␣
      issus␣des␣diff rentes␣m thodes␣de␣
      v rification."@fr;
  rdfs:label "R sultat␣de␣V rification␣(Interne)"
      @fr .

:SocialMediaPlatform a owl:Class;
  rdfs:subClassOf :Source;
  rdfs:label "Plateforme␣de␣M dia␣Social"@fr .

:Source a owl:Class;
  rdfs:comment "Repr sente␣une␣entit ␣(site␣web,␣
      organisation,␣personne)␣d'o ␣provient␣l'
      information␣originale␣ou␣la␣preuve."@fr;
  rdfs:label "Source"@fr .

:SupportingEvidence a owl:Class;
  rdfs:subClassOf :Evidence;
  rdfs:label "Preuve␣ ␣l'appui"@fr .

:SystemeExterne a owl:Class;
  rdfs:comment "Repr sente␣une␣source␣de␣donn es␣
      ou␣un␣service␣externe␣utilis ␣pendant␣le␣
      processus␣de␣v rification␣(API,␣base␣de␣
      donn es)."@fr;
  rdfs:label "Syst me␣Externe"@fr .

:User a owl:Class;
  rdfs:comment "Repr sente␣une␣personne␣
      interagissant␣avec␣le␣syst me␣de␣
      v rification."@fr;
  rdfs:label "Utilisateur"@fr .

:VerificationCriterion a owl:Class;
  rdfs:comment "Aspect␣sp cifique␣ valu ␣lors␣de␣
      la␣v rification␣(ex:␣r putation␣de␣la␣
      source,␣coh rence)."@fr;
  rdfs:label "Crit re␣de␣V rification"@fr .

:VerificationMethod a owl:Class;
  rdfs:comment "Repr sente␣une␣approche␣(r gle,␣
      mod le␣IA)␣utilis e␣pour␣ valuer ␣la␣
      cr dibilit ."@fr;
  rdfs:label "M thode␣de␣V rification"@fr .

:Criteria_AuthorExpertise a owl:NamedIndividual, :
    VerificationCriterion;
  rdfs:label "Expertise␣de␣l'auteur"@fr .

:Criteria_CoherenceAnalysis a owl:NamedIndividual, :
    VerificationCriterion;
  rdfs:label "Analyse␣de␣la␣coh rence"@fr .

:Criteria_CrossReferencing a owl:NamedIndividual, :
    VerificationCriterion;
  rdfs:label "R f rences␣crois es"@fr .

:Criteria_FactCheckDB a owl:NamedIndividual, :
    VerificationCriterion;
  rdfs:label "Consultation␣base␣de␣donn es␣Fact-
      Check"@fr .

:Criteria_SourceReputation a owl:NamedIndividual, :
    VerificationCriterion;
  rdfs:label "R putation␣de␣la␣source"@fr .

:Criteria_ToneAnalysis a owl:NamedIndividual, :
    VerificationCriterion;
  rdfs:label "Analyse␣du␣ton␣(ex:␣neutre,␣biais)"
      @fr .

_:genid60 owl:maxCardinality "1"^^xsd:
    nonNegativeInteger .

_:genid61 a owl:AllDisjointClasses;
  owl:members _:genid66 .

_:genid66 a rdf:List;
  rdf:first :AcademicJournal;
  rdf:rest _:genid65 .

_:genid65 a rdf:List;
  rdf:first :FactCheckingOrganization;
  rdf:rest _:genid64 .

_:genid64 a rdf:List;
  rdf:first :NewsWebsite;
  rdf:rest _:genid63 .

_:genid63 a rdf:List;
  rdf:first :PersonalBlog;
  rdf:rest _:genid62 .

_:genid62 a rdf:List;
  rdf:first :SocialMediaPlatform;
  rdf:rest rdf:nil .

_:genid67 a owl:AllDisjointClasses;
  owl:members _:genid70 .

_:genid70 a rdf:List;
  rdf:first :ApiLLM;
  rdf:rest _:genid69 .

_:genid69 a rdf:List;
  rdf:first :BaseDeFaits;
  rdf:rest _:genid68 .

_:genid68 a rdf:List;
  rdf:first :MoteurRecherche;
  rdf:rest rdf:nil .

_:genid71 a owl:AllDisjointClasses;
  owl:members _:genid74 .

_:genid74 a rdf:List;
```

```
  rdf:first :InformationFaibleCredibilite;
  rdf:rest _:genid73 .

_:genid73 a rdf:List;
  rdf:first :InformationHauteCredibilite;
  rdf:rest _:genid72 .

_:genid72 a rdf:List;
  rdf:first :InformationMoyenneCredibilite;
  rdf:rest rdf:nil .

_:genid75 a owl:AllDisjointClasses;
  owl:members _:genid79 .

_:genid79 a rdf:List;
  rdf:first :Niveau_Bas;
  rdf:rest _:genid78 .

_:genid78 a rdf:List;
  rdf:first :Niveau_Haut;
  rdf:rest _:genid77 .

_:genid77 a rdf:List;
  rdf:first :Niveau_Moyen;
  rdf:rest _:genid76 .

_:genid76 a rdf:List;
  rdf:first :Niveau_NonVerifie;
  rdf:rest rdf:nil .
```