Hybrid System for Verifying the Credibility of Information Sources: An Ontology

DOMINIQUE S. LOYER[1]

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

May 16, 2025

This document accompanies the OWL 2 DL ontology created to model the domain of evaluating the credibility of information on the Internet, based on a prior UML modeling Loyer2025Rapport and inspired by recognized structures from a grant management ontology Loyer2025OntoSubv. The ontology, developed in Turtle syntax and designed for use with tools such as Protégé Protege2025, aims to capture the essential concepts (information, source, query, report, evaluation methods, criteria, credibility levels), their interrelationships, and some fundamental axioms for consistency and possible inference. This text provides an in-depth explanation of the modeled domain, an inventory of classes and properties, as well as an evaluation of the modeling choices regarding the types of properties, restrictions, and OWL axioms chosen, including how the structure facilitates the categorization of information according to its inferred credibility level. The objective is to record the structure and reasoning of the ontology created for the credibility assessment system.

1. INTRODUCTION

This report presents the creation and modeling of an OWL 2 ontology W3C2012OWL2 for the field of evaluating the reliability of information on the Internet. This project is related from an initial modeling carried out with UML Loyer2025Rapport, Fowler2003, Larman2004. The main challenge discussed is the propagation of misinformation and the difficulty for users to distinguish credible information Zhou2020, Viviani2017, Metzger2010. The purpose of the ontology is to offer a formal semantic representation for a credibility assessment system, by capturing the essential entities (information provided, sources, user queries, evaluation reports, analytical methods, criteria, evidence) and their relationships.

The ontology was created with the OWL 2 DL language and written in Turtle syntax TurtleSpec. The Protégé application Protege2025 is intended for loading, visualizing, and exploiting the ontology. The methodology is based on the fundamental principles of ontology creation NoyMcGuinness2001 and incorporates structural elements drawn from a previous ontology concerning grant management Loyer2025OntoSubv, particularly for the modeling of criteria-based evaluation.

This report is structured[1] as follows: Section 3 presents a detailed description of the modeled domain, including the classes[2] and properties added or adapted. Section 4 explains in detail the specific modeling choices by emphasizing the types and axioms[3] of properties and classes, as well as the restrictions used. Section 5 discusses how the ontology supports the classification of information based on credibility. Section 7 concludes the report. References are provided at the end.

2. STATE OF THE ART

In the field of information credibility verification, several approaches and systems have been developed. These efforts often combine machine learning, natural language processing, and knowledge representation techniques to help users distinguish reliable information from misleading content Oshikawa2020, Sharma2019, deSouza2020. Hybrid approaches, combining expert rules with machine learning models, have become common to leverage the strengths of each method Ahmed2024. Rules can capture explicit indicators of credibility or misinformation, while AI models can identify more subtle patterns and adapt to new forms of misleading content, including those generated by language models Loth2024, ChenShu2023. The use of ontologies and the Semantic Web for information verification has also been explored. Ontologies allow the formalization of key concepts, their relationships, and domain rules, thereby facilitating reasoning and inference about the credibility of sources and information Pescuma2025. Initiatives like Schema.org with its ‘ClaimReview‘ vocabulary aim to structure fact-checking information on the web ClaimReviewSchema. Recent work has also focused on the explainability of information verification systems, seeking to provide users with clear and understandable justifications for credibility assessments he2023debunkingdisinformationrevolutionizingtruth. This is particularly important for fostering trust in these systems and for helping users develop their own critical thinking skills.

[1] See the outline at the end of the document
[2] See Appendix II of the Turtle code from line 141
[3] See the rules of Appendix II of the Turtle code from line 266 to 1030

3. DETAILED DESCRIPTION OF THE MODELED DOMAIN

The modeled domain concerns the process of verifying the credibility of information[4] found online (e.g., a news article, a blog post, a claim). The target system, as described in the UML modeling Loyer2025Rapport, aims to take information submitted by a user[5], analyze it using a hybrid approach (logical rules and AI/NLP models), consult external sources (search engines, verified fact databases, LLM APIs), and produce an evaluation report including a credibility level and justifications.

The ontology must capture the following elements:

- **The query process:** A user submits information, which triggers an evaluation query. This query goes through different states (submitted, processing, analyzed, report generated, error) and results in a report.
- **The information and its source:** The submitted information (text, URL) has a presumed original source and potentially an identified author.
- **The actors:** The average user who submits the query, and the expert who configures and maintains the system (rules, models).
- **The hybrid approach:** The system uses predefined verification rules and AI/NLP models to analyze the information.
- **The evaluation criteria:** The evaluation is based on specific criteria (e.g., source reputation, text coherence, presence of bias, tone analysis).
- **The intermediate results:** Each rule or model produces a result that contributes to the evaluation of one or more criteria.
- **The external sources:** The system interacts with external APIs to collect data and evidence.
- **The evidence:** External information (corroborating or refuting articles, fact database entries) is collected as evidence.
- **The final report:** The report synthesizes the analyses, presents the evidence, details the results by criterion, and assigns an overall credibility level (e.g., High, Medium, Low).
- **The classification:** The ontology must allow the inference of an information's classification (e.g., InformationHauteCredibilite) based on the credibility level determined in its report.

A. Modeled Classes

[6]

To represent this domain, the following classes have been defined (non-exhaustive list, see complete ontology for details):

Main Classes of the Process:

- RequeteEvaluation: Represents a unique verification request.
- InformationSoumise: The information (text/URL) provided by the user.
- RapportEvaluation: The final result produced for a query.
- User: The user initiating the request.
- Expert: Sub-class of User, responsible for configuration.

Classes related to Information and its Provenance:

- Source: The origin of the information or evidence (website, organization, etc.). Sub-classes: NewsWebsite, SocialMediaPlatform, AcademicJournal, PersonalBlog, FactCheckingOrganization.
- Author: The presumed author of the information.

Classes related to Evaluation and Methods:

- VerificationMethod: Parent class for analysis methods.
- RegleVerification: A specific logical rule.
- ModeleIA: A specific AI/NLP model.
- VerificationCriterion: An evaluation criterion (e.g., reputation, coherence). Predefined individuals: Criteria_SourceReputation, Criteria_AuthorExpertise, etc.
- ResultatVerification: Parent class for intermediate results.
- ResultatRegle: Result of applying a rule.
- ResultatNLP: Result of applying an AI model.
- ResultatCritere: Aggregated result for a specific criterion (inspired by NoteAttribuee in Loyer2025OntoSubv).
- CredibilityLevel: The final credibility level (e.g., High, Medium, Low). Predefined individuals: Niveau_Haut, Niveau_Moyen, etc.

Classes related to External Data and Evidence:

- SystemeExterne: An external data source (API, DB). Sub-classes: MoteurRecherche, ApiLLM, BaseDeFaits.
- Evidence: External evidence used in the evaluation. Sub-classes: SupportingEvidence, RefutingEvidence.
- InfoSourceAnalyse: Details about an analyzed source in the report.

Definitional Classes (for classification):

- InformationVerifiee: Information that has been the subject of a report.
- InformationHauteCredibilite, InformationMoyenneCredibilite, InformationFaibleCredibilite: Classes defined by logical equivalence (`owl:equivalentClass`) based on the CredibilityLevel assigned in the associated report.

[4] See verification criteria Fig. 9
[5] See the interface in Fig. 3
[6] See Fig 1

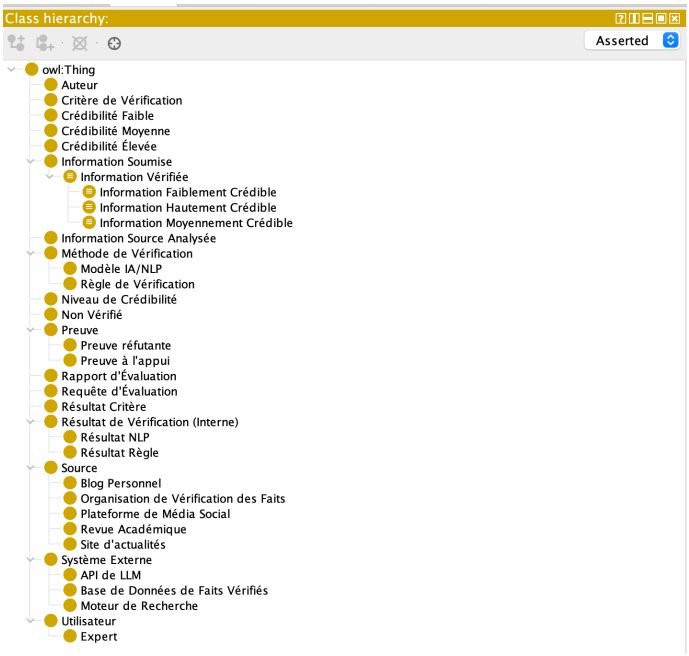

Fig. 1. View of classes in Protégé

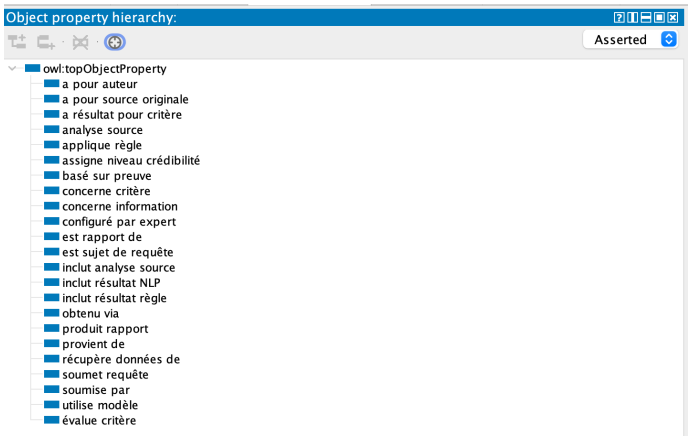

Fig. 2. View of object properties in Protégé

Fig. 3. Simplistic user interaction interface

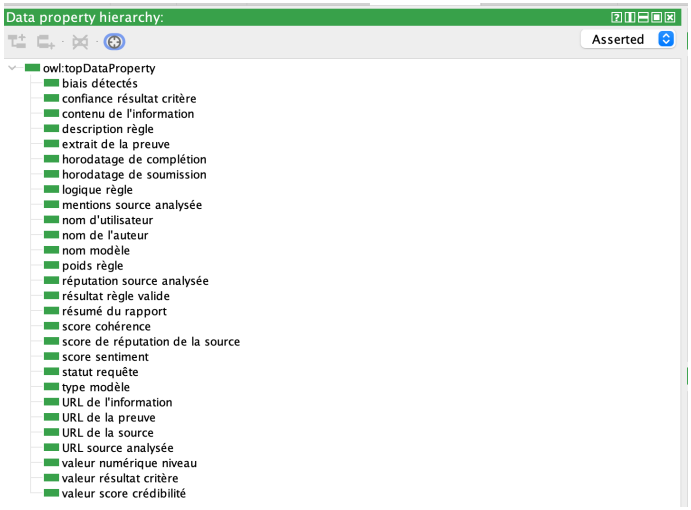

Fig. 4. View of data properties in Protégé

B. Modeled Properties

7

The relationships between these classes and their attributes are modeled via object and data properties.

Key Object Properties:

- concernsInformation, submittedBy, producesReport (and their inverses) to link RequeteEvaluation, InformationSoumise, User, and RapportEvaluation.
- hasOriginalSource, hasAuthor to characterize InformationSoumise.
- includesRuleResult, includesNLPResult, includesSourceAnalysis, basedOnEvidence, assignsCredibilityLevel to structure RapportEvaluation.
- appliesRule, usesModel to link results to their methods.
- evaluatesCriterion to link methods (RegleVerification, ModeleIA) to the criteria they evaluate.
- hasCriterionResult, concernsCriterion, obtainedVia to detail results by criterion (inspired by Loyer2025OntoSubv).
- originatesFrom to link Evidence to its Source.
- configuredByExpert, fetchesDataFrom.

Key Data Properties:

- requestStatus, submissionTimestamp (on RequeteEvaluation).
- informationContent, informationURL (on InformationSoumise).
- credibilityScoreValue, reportSummary, completionTimestamp (on RapportEvaluation).
- ruleLogic, ruleWeight, ruleDescription (on RegleVerification).
- ruleResultValid (on ResultatRegle).
- modelName, modelType (on ModeleIA).
- sentimentScore, coherenceScore, detectedBiases (on ResultatNLP).
- criterionResultValue, criterionResultConfidence (on ResultatCritere).
- sourceURL, sourceReputationScore (on Source).
- evidenceURL, evidenceSnippet (on Evidence).
- credibilityLevelValue (on CredibilityLevel).

4. MODELING CHOICES

This section details the choices made to represent the key elements of the domain in OWL 2 W3C2012OWL2, using RDF and RDFS vocabularies, and justifies these choices with respect to the ontology's objectives and best practices NoyMcGuinness2001.

[7] See Fig 2

A. Property Types (Characteristics)

8

Several properties have been declared with specific characteristics to enrich semantics and enable inferences or consistency checks:

- **Functional (`owl:FunctionalProperty`):** Ensures that an individual[9] has at most one value for the property. Applied to properties where a single relation is expected, for example:
  - concernsInformation (a query concerns only one piece of information),
  - submittedBy (a query is submitted by only one user),
  - producesReport (a query produces only one report),
  - assignsCredibilityLevel (a report assigns only one level),
  - appliesRule / usesModel (a specific result comes from only one rule/model),
  - concernsCriterion (a criterion result concerns only one criterion),
  - credibilityScoreValue, credibilityLevelValue, submissionTimestamp, completionTimestamp, requestStatus, sourceURL.
- **Inverse Functional (`owl:InverseFunctionalProperty`):** Ensures that a given value is associated with only one subject via this property. Applied to isReportOf (a report corresponds to only one query).[10]
- **Inverses (`owl:inverseOf`):** Explicitly defined for pairs of reciprocal properties (e.g., concernsInformation / isSubjectOfRequest, submittedBy / submitsRequest, producesReport / isReportOf). This allows the reasoner to infer a relation if its inverse is known.
- **Other Characteristics:** No properties were declared transitive, symmetric, etc., as this did not seem relevant for the described domain.

B. Property Restrictions

Restrictions (`owl:Restriction`) are used to define classes more precisely, often by constraining the values or number of values for a given property:

- **Cardinality Restrictions (`owl:cardinality`, `owl:minCardinality`, `owl:maxCardinality`):**
  - Used on RequeteEvaluation to require exactly 1 concernsInformation and 1 submittedBy, and at most 1 producesReport (allowing for ongoing queries).
  - Used on RapportEvaluation to require exactly 1 assignsCredibilityLevel.
  - Used on ResultatRegle, ResultatNLP, ResultatCritere, InfoSourceAnalyse to link these results to their unique source entity (rule, model, criterion, source).
  - Used on RegleVerification and ModeleIA to indicate that they must evaluate (evaluatesCriterion) at least 1 VerificationCriterion.
  - Used on ResultatCritere to indicate that it must be obtained via (obtainedVia) at least 1 ResultatRegle or ResultatNLP.

[8] See Fig. 4

[9] See instantiations in Appendix II of the Turtle code from line 224

[10] See Fig. 10 for the inverse functional characteristic for isReportOf and producesReport

  - Central to the `owl:equivalentClass` axioms for credibility classes (InformationHauteCredibilite, etc.) to filter on the value of the credibility level.

- **Value Restrictions (`owl:allValuesFrom`, `owl:someValuesFrom`, `owl:hasValue`):**
  - `owl:someValuesFrom` is used in `owl:equivalentClass` axioms to express existential conditions (e.g., information is verified if it is the subject of a query that produces a report).
  - `owl:hasValue` is used in the innermost restrictions of credibility classification axioms to specify that the value of the assignsCredibilityLevel property must be a specific individual (e.g., Niveau_Haut, Niveau_Moyen, Niveau_Bas).
  - `owl:allValuesFrom` is used in `rdfs:subClassOf` axioms linking definitional classes (e.g., InformationHauteCredibilite) to the corresponding value of assignsCredibilityLevel in the associated report.

C. Class Axioms

[11]

- **Hierarchy (`rdfs:subClassOf`):** Defines the basic taxonomic structure (e.g., Expert is a User, NewsWebsite is a Source, ResultatRegle is a ResultatVerification).
- **Disjointness (`owl:AllDisjointClasses`):** Used to declare that certain classes are mutually exclusive:
  - The different types of Source (e.g., NewsWebsite, SocialMediaPlatform).
  - The different types of SystemeExterne.
  - The types of Evidence (SupportingEvidence, RefutingEvidence).
  - The types of internal results (ResultatRegle, ResultatNLP).
  - The different CredibilityLevel (e.g., Niveau_Haut, Niveau_Moyen).
  - The information classes defined by their credibility (InformationHauteCredibilite, InformationMoyenneCredibilite, InformationFaibleCredibilite). This ensures that information cannot be classified into several of these categories simultaneously via inference.
- **Boolean Combinations (`owl:intersectionOf`, `owl:unionOf`, `owl:complementOf`):**
  - `owl:intersectionOf` is essential for the `owl:equivalentClass` definitions of credibility classes, combining the condition of being InformationVerifiee with the restriction on the assigned credibility level, and potentially complements for exclusivity.
  - `owl:unionOf` is used in the domains/ranges of some properties to allow multiple types (e.g., the domain of configuredByExpert is the union of RegleVerification, ModeleIA, VerificationCriterion).
  - `owl:complementOf` is used in the definitions of InformationMoyenneCredibilite and InformationFaibleCredibilite classes to explicitly exclude information already belonging to a higher credibility category.
- **Equivalent Classes (`owl:equivalentClass`):** At the heart of the automatic credibility classification. The classes InformationHauteCredibilite, InformationMoyenneCredibilite, InformationFaibleCredibilite are defined as logically equivalent to complex descriptions based on the CredibilityLevel (Niveau_Haut, etc.) assigned via the RapportEvaluation associated with the RequeteEvaluation concerning the InformationSoumise. This allows the OWL reasoner to automatically infer the membership of information in one of these classes if it satisfies the conditions.[12]

D. Property Axioms

- **Inverse (`owl:inverseOf`):** The main property axiom used, to link pairs of reciprocal properties.
- **Other:** No axioms for property disjointness (`owl:propertyDisjointWith`) or property chains (`owl:propertyChainAxiom`) have been defined, as this did not appear necessary for the current model.

5. CREDIBILITY-BASED CLASSIFICATION

A key objective of the ontology is to enable the automatic classification of an instance of InformationSoumise based on the credibility level determined by the system. This mechanism is inspired by the classification of grant proposals seen in Loyer2025OntoSubv.

A. Classification Logic

The classification is based on the `owl:equivalentClass` axioms defined for the InformationHauteCredibilite, InformationMoyenneCredibilite, and InformationFaibleCredibilite classes. The logic is as follows:

1. An InformationSoumise is first considered InformationVerifiee if it is linked (isSubjectOfRequest) to a RequeteEvaluation that has produced (producesReport) an RapportEvaluation.
2. Then, to be classified as InformationHauteCredibilite, an InformationVerifiee must be linked (via isSubjectOfRequest then producesReport) to a RapportEvaluation that assigns (assignsCredibilityLevel) the individual Niveau_Haut.
3. To be classified as InformationMoyenneCredibilite, it must be an InformationVerifiee, be linked to a report assigning Niveau_Moyen, AND not be an InformationHauteCredibilite (via `owl:complementOf`).
4. To be classified as InformationFaibleCredibilite, it must be an InformationVerifiee, be linked to a report assigning Niveau_Bas, AND not be either InformationHauteCredibilite or InformationMoyenneCredibilite.

This structure, using `owl:equivalentClass` and `owl:complementOf`, allows the OWL reasoner to infer the appropriate credibility class for a given piece of information, provided that the instances (RequeteEvaluation, RapportEvaluation) and the relationships (assignsCredibilityLevel) are correctly asserted.

B. Detailed Modeling of Results by Criterion

The introduction of the ResultatCritere class and the associated properties (hasCriterionResult, concernsCriterion, obtainedVia, criterionResultValue) allows for finer modeling, analogous to the management of grades by criterion in the grant ontology Loyer2025OntoSubv.

[11] See the rules in Appendix II line 266 to 1030

[12] See e.g., boolean combination intersectionOf in Fig. 11

- Each RapportEvaluation can have several instances of ResultatCritere (via hasCriterionResult).
- Each ResultatCritere concerns a specific VerificationCriterion (via concernsCriterion).
- It is obtained via one or more ResultatRegle or ResultatNLP (via obtainedVia).
- It has a value (criterionResultValue, e.g., "Weak Reputation", "High Coherence", "Political Bias Detected") and potentially a confidence (criterionResultConfidence).

Although the current ontology does not directly use ResultatCritere for the final classification (which is based on the overall CredibilityLevel), this structure provides the basis for more complex future reasoning or for the detailed explanation of the final score in the report. For example, one could define information classes based on the presence of specific results for certain criteria (e.g., "Information with Weak Source Reputation").

C. Testing by Inference

[13]

To verify the operation of this classification in Protégé:

1. Create an instance of InformationSoumise (e.g., `infoTestHaut`).
2. Create an instance of RequeteEvaluation (e.g., `reqTestHaut`) and link it to `infoTestHaut` via concernsInformation.
3. Create an instance of RapportEvaluation (e.g., `rapTestHaut`) and link it to `reqTestHaut` via producesReport.
4. Assert the assignsCredibilityLevel property on `rapTestHaut` with the value Niveau_Haut.
5. Launch a reasoner (e.g., HermiT).
6. Verify that the reasoner infers that `infoTestHaut` is of type InformationHauteCredibilite (in addition to InformationSoumise and InformationVerifiee).
7. Repeat for the other levels (Niveau_Moyen, Niveau_Bas) ensuring that the classifications are exclusive.

This procedure allows validation that the `owl:equivalentClass` axioms function as expected for classification based on the final credibility level.[14]

6. RESULTS AND LIMITATIONS

The developed ontology makes it possible to formally represent the domain of online information credibility verification. The main result is an OWL 2 DL structure that captures the essential concepts and their relationships, thus allowing for the modeling of the credibility assessment process through a hybrid approach. The classification section demonstrates the ontology's ability to infer the credibility level of submitted information based on the level assigned in the evaluation report. This functionality is a direct result of using OWL equivalence axioms and restrictions. The modeling of results by criterion provides a basis for more detailed analysis and potentially for more complex inference in the future. However, the current ontology has some limitations. The detail of the verification rules modeling and the outputs of the AI/NLP models remain abstract. Although the hybrid approach is conceptually represented, the exact logic of combining the results is not encoded at the ontological level. Integration with existing external ontologies (for example, for source types or thematic domains) is limited and could enrich the model's semantics. Furthermore, the ontology does not explicitly specify the logic that allows deriving the value of a result by criterion from the results of rules and AI models, nor the aggregation logic to determine the final credibility level. These aspects could be addressed by future extensions using semantic rule languages.

7. CONCLUSION AND FUTURE WORK

The elaborated OWL 2 DL ontology[15] provides a formal semantic representation for the domain of information credibility verification, aligning with a prior UML modeling Loyer2025Rapport and integrating evaluation structures inspired by a grant management ontology Loyer2025OntoSubv. It captures the key entities of the process (query, information, source, report), the actors (user, expert), the hybrid analysis methods (rules, AI), the evaluation criteria, and the associated results, including the final credibility levels.

The use of `owl:equivalentClass` axioms, combined with value restrictions and complements, allows for the automatic classification of verified information into credibility categories (High, Medium, Low) based on the credibility level assigned in the generated report. The detailed modeling of results by criterion via the ResultatCritere class offers fine granularity for analysis and potential explanations.

The ontology is designed to be usable with standard tools such as Protégé Protege2025, allowing for exploration, instantiation, and reasoning.

Perspectives and Future Work: Several areas for improvement can be considered:

- **Refinement of Criteria and Rules:** Detail further the types of VerificationCriterion and potentially model the logic of certain RegleVerification if recurring and formalizable patterns exist.
- **Modeling of AI Outputs:** Represent more finely the types of biases or the levels of sentiment/coherence as individuals rather than simple character strings in ResultatNLP.
- **Integration of External Ontologies:** Align the ontology with existing vocabularies for source types (e.g., Schema.org), thematic domains, or bias types.
- **Advanced Reasoning:** Explore the use of SWRL or other rule languages to encode the logic that determines the value of an ResultatCritere from ResultatRegle/ResultatNLP, or even the logic that determines the final CredibilityLevel from ResultatCritere. This would increase complexity but allow for richer inferences.
- **Explainability:** Use the structure of the ontology (links between report, criterion results, rule/nlp results, evidence) to generate more structured and understandable explanations for the end user.
- **Uncertainty Management:** Integrate mechanisms to represent the confidence or uncertainty associated with the different stages of the evaluation (e.g., via data properties like criterionResultConfidence or by exploring probabilistic extensions of OWL).

In conclusion, this ontology provides a solid semantic foundation for the development and explanation of an information credibility assessment system, while offering clear avenues for future enhancements.

[13] See Fig 5 to 8

[14] See changes made during these tests in Figure 12

[15] See the graph taxonomy in Fig. 13

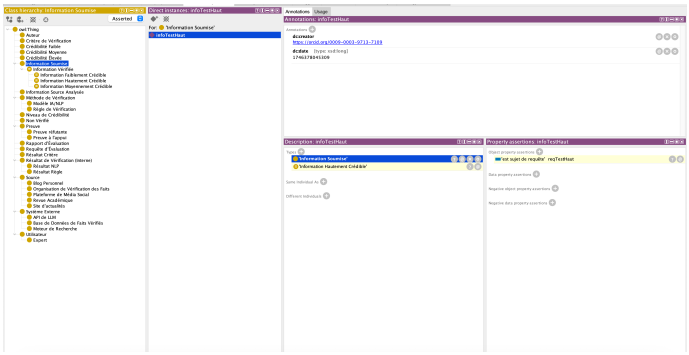

Fig. 5. Inference on analyzed information

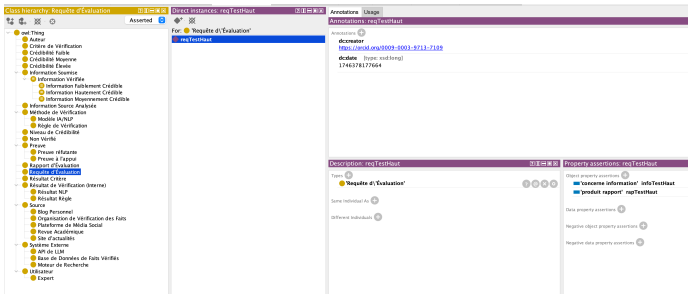

Fig. 6. Query and report linkage

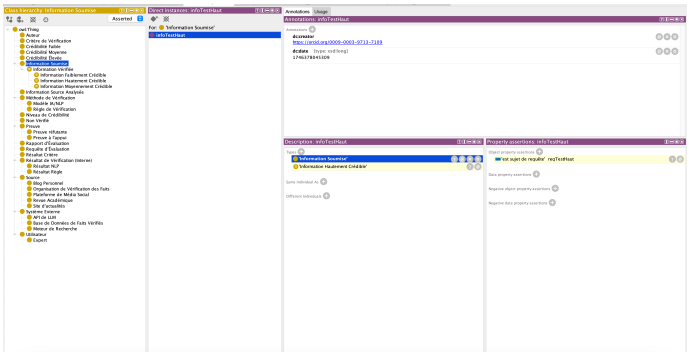

Fig. 7. Test inference with other credibility

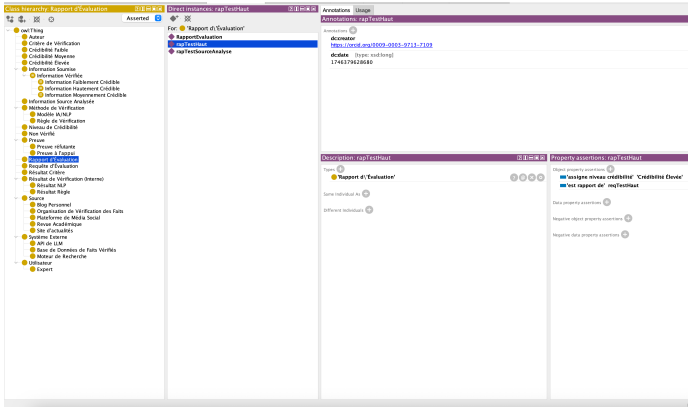

Fig. 8. Test report with high credibility

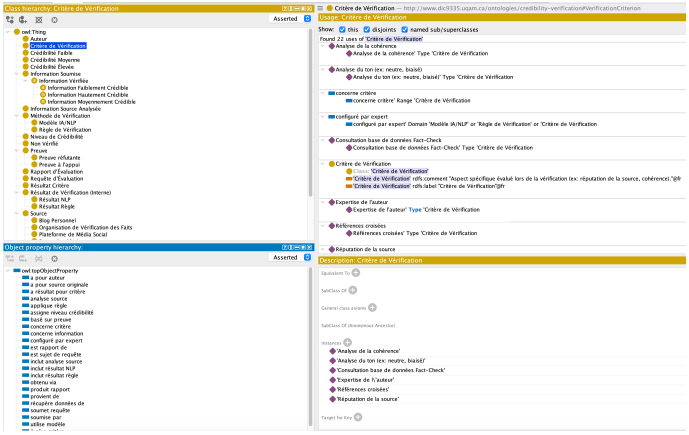

Fig. 9. The verification criteria

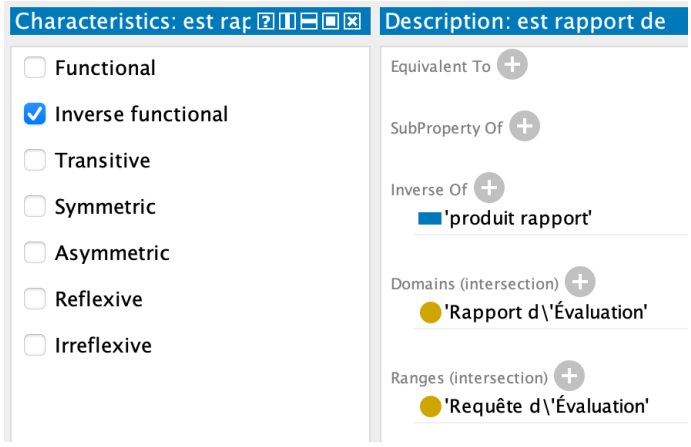

Fig. 10. e.g., of inverse functional characteristic for isReportOf

Fig. 11. e.g., of intersections of the domains AI/NLP Models OR Verification Rules OR Verification Criteria with co-domain Expert

```
:InformationSoumise a owl:Class;
  rdfs:comment "Représente l'unité d'information (texte, URL) telle que soumise pour vérification."@fr;
  rdfs:label "Information Soumise"@fr .

:InformationVerifiee a owl:Class ;
    owl:equivalentClass [
        a owl:Restriction ;
        owl:onProperty :isSubjectOfRequest ;
        owl:someValuesFrom [
            a owl:Class ;
            owl:intersectionOf (
                :RequeteEvaluation
                [ a owl:Restriction ;
                  owl:onProperty :producesReport ;
                  owl:someValuesFrom :RapportEvaluation
                ]
            )
        ]
    ] ;
    rdfs:label "Information Vérifiée"@fr .
```

Fig. 12. Changes made following tests

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. Preprint, ResearchGate.
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. Beckett, D., Berners-Lee, T., Prud'hommeaux, E., and Carothers, G. (2014). Rdf 1.1 turtle: Terse rdf triple language. W3C Recommendation.
6. 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.
7. Chen, K. and Shu, K. (2023). Combating misinformation in the age of llms: Opportunities and challenges.
8. Chen, W.-F. and Oh, A. (2020). Detecting media bias using gaussian mixture models: An unsupervised approach.
9. 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.
10. Fowler, M. (2003). *UML Distilled: A Brief Guide to the Standard Object Modeling Language*. Addison-Wesley, 3rd edition.
11. 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.
12. 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.
13. Google (s.d.a). Custom search json api. Google Developers Documentation. Accessed in April 2025.
14. Google (s.d.b). Fact check tools api. Google Developers Documentation. Accessed in April 2025.
15. 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.
16. He, L., Hu, S., and Pei, A. (2023). Debunking disinformation: Revolutionizing truth with nlp in fake news detection.
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, 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.
21. Lounis, H. (2025). Modélisation des problèmes complexes en sciences cognitives. Présentations et diapositives du cours DIC-9251, UQAM.
22. Loyer, D. S. (2025a). Project 1: modeling a system allowing for the explainability of information sources. Project Report DIC-9251, UQAM.
23. Loyer, D. S. (2025b). Report: Ontology of the public research funding domain. Project report, UQAM. Presented as part of the course DIC-9335.
24. Menzner, P. and Leidner, J. L. (2024). Biasscanner: Detecting biased statements in news articles.
25. 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.
26. Microsoft (s.d.). Bing search apis documentation. Microsoft Azure Documentation. Accessed in April 2025.
27. Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space.
28. 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.
29. Noy, N. F. and McGuinness, D. L. (2001). Ontology development 101: A guide to creating your first ontology. Technical Report KSL-01-05 and SMI-2001-0880, Stanford Knowledge Systems Laboratory.
30. 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.
31. Oshikawa, R., Qian, J., and Wang, W. Y. (2020). A survey on natural language processing for fake news detection. v3.
32. 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.
33. Pescuma, V. N., Osborne, F., Recupero, D. R., and Motta, E. (2025). Source Credibility Assessment: A Comprehensive Survey. *International Journal of Interactive Multimedia and Artificial Intelligence*, TODO:TODO. Advance online publication.
34. Prabhakaran, V., Hutchinson, B., and Mitchell, M. (2019). Perturbation sensitivity analysis to detect unintended model biases.
35. 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.
36. Rieh, S. Y. (2010). Credibility assessment of online information in context. *Information Research*, 15(3). paper 445.
37. 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.
38. Schema.org Community (s.d.). Claimreview - schema.org type. Schema.org Documentation. Accessed in April 2025.
39. Shah, B., Shah, D., and Attar, V. (2025). Decoding news bias: Multi bias detection in news articles. v1.
40. 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.
41. The Protégé Project (2025). Protégé ontology editor. https://protege.stanford.edu/. (Accessed in April 2025).
42. Thibault, C., Tian, J.-J., Peloquin-Skulski, G., Curtis, T. L., Zhou, J., Laflamme, F., Guan, Y., Rabbany, R., Godbout, J.-F., and Sinha, K. (2025). A guide to misinformation detection data and evaluation. v2.
43. 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.
44. W3C Credibility Community Group (2018). Credibility signals. W3C Community Group Draft Report.
45. W3C OWL Working Group (2012). Owl 2 web ontology language document overview (second edition). W3C Recommendation.
46. 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.
47. Zhou, X. and Zafarani, R. (2020). A survey of fake news: Fundamental theories, detection methods, and opportunities. In *ACM Computing Surveys (CSUR)*, volume 53, pages 1–40.

CONTENTS

8. APPENDICE 0 TAXONOMY'S ONTOLOGY

9. APPENDICE I (PYTHON CODE FOR THE HYBRID SYS.-PROTOTYPE)

```python
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
# 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", grouped_entities=True) # grouped_entities est
    souvent utile

# 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.
explainer = LimeTextExplainer(class_names=['NEGATIVE', 'POSITIVE']) # Ajuster si le
     mod le a d'autres classes

# --- 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"[Simulation] 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"[Simulation]  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"[Simulation] 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):
        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'] = [
        {'title': 'Related Story A', 'url': 'http://example.com/a'},
        {'title': 'Related Story B', 'url': 'http://example.com/b'}
    ]
    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)
```

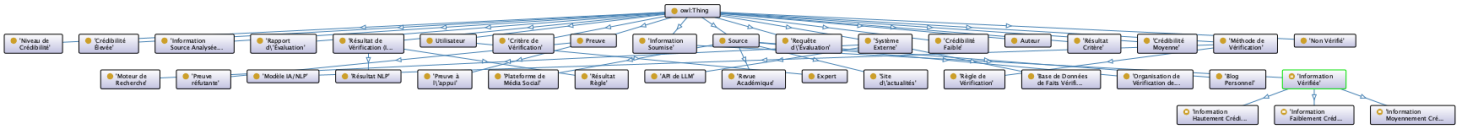

Fig. 13. Ontology's graph

```
        # 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
                predictions = self.sentiment_pipeline(valid_texts)
                pred_idx = 0
                for original_text in processed_texts:
                    if original_text: # Si le texte original n' tait  pas vide
                        apr s preprocess
                        pred = predictions[pred_idx]
                        # Assurer que la sortie est toujours [prob_neg, prob_pos]
                        if pred['label'] == 'POSITIVE':
                            probabilities.append([1 - pred['score'], pred['score'
                                ]])
                        else: # NEGATIVE or other label mapped to negative
                            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 = self.sentiment_pipeline(text)[0]
            results['sentiment'] = main_prediction

            # 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
            simulated_bias_score = torch.softmax(logits, dim=1)[0][0].item() #
                Exemple
            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 considrablement.
        Le score va de 0 (peu cr dible)    1 (tr s  cr dible).
        """
        score = 0.5 # Score de base neutre
        weight_sum = 1.0 # Pour normaliser les poids ajout s/soustraits
        score_adjustment = 0.0

        # --- Pond rations (Exemples -    AJUSTER ABSOLUMENT) ---
        WEIGHT_REPUTATION = 0.3
        WEIGHT_AGE = 0.05
        WEIGHT_CERTAINTY = 0.1
        WEIGHT_DOUBT = 0.15
        WEIGHT_SENSATIONALISM = 0.1
        WEIGHT_NEGATIVE_SENTIMENT = 0.05
        WEIGHT_BIAS = 0.15
        WEIGHT_COHERENCE = 0.05 # Faible car simul

        # Facteurs bas s sur les r gles
        if rule_results['source_analysis']['reputation'] == 'High':
            score_adjustment += WEIGHT_REPUTATION
            weight_sum += WEIGHT_REPUTATION
        elif rule_results['source_analysis']['reputation'] == 'Low':
            score_adjustment -= WEIGHT_REPUTATION
            weight_sum += WEIGHT_REPUTATION

        domain_age = rule_results['source_analysis'].get('domain_age_days')
        # *** CORRECTION ICI AUSSI *** V rifier si domain_age n'est pas None
        if domain_age is not None:
            if domain_age > 365 * 2: # Ex: > 2 ans
                score_adjustment += WEIGHT_AGE
                weight_sum += WEIGHT_AGE
            elif domain_age < 90: # Ex: < 3 mois
                score_adjustment -= WEIGHT_AGE
                weight_sum += WEIGHT_AGE


        if rule_results['linguistic_markers']['certainty'] > 0 and rule_results['
            linguistic_markers']['doubt'] == 0:
            score_adjustment += WEIGHT_CERTAINTY * rule_results['
                linguistic_markers']['certainty'] # Plus de certitude = plus de
                poids
            weight_sum += WEIGHT_CERTAINTY * rule_results['linguistic_markers']['
                certainty']
        elif rule_results['linguistic_markers']['doubt'] > 0:
            score_adjustment -= WEIGHT_DOUBT * rule_results['linguistic_markers']['
                doubt'] # P naliser plus si plusieurs mots
            weight_sum += WEIGHT_DOUBT * rule_results['linguistic_markers']['doubt
                ']

        if rule_results['linguistic_markers']['sensationalism'] > 0:
            sensationalism_penalty = min(WEIGHT_SENSATIONALISM * rule_results['
                linguistic_markers']['sensationalism'], WEIGHT_SENSATIONALISM *
                3)
            score_adjustment -= sensationalism_penalty
            weight_sum += sensationalism_penalty

        # Facteurs bas s sur le NLP
        if nlp_results.get('sentiment'): # V rifier que la cl  'sentiment' existe
            if nlp_results['sentiment']['label'] == 'NEGATIVE' and nlp_results['
                sentiment']['score'] > 0.85:
                score_adjustment -= WEIGHT_NEGATIVE_SENTIMENT
                weight_sum += WEIGHT_NEGATIVE_SENTIMENT

        if nlp_results.get('bias_analysis') and 'Flagged' in nlp_results['
            bias_analysis'].get('label',''):
            bias_score_value = nlp_results['bias_analysis'].get('score')
            if bias_score_value is not None:
                bias_impact = WEIGHT_BIAS * ((bias_score_value - 0.5) * 2) #
                    Normaliser le score de biais (0.5->0, 1.0->1)
                score_adjustment -= bias_impact # Soustraire l'impact du biais
                weight_sum += WEIGHT_BIAS # Ajouter le poids du facteur biais

        if nlp_results.get('coherence_score') is not None:
            coherence_adjustment = (nlp_results['coherence_score'] - 0.5) *
                WEIGHT_COHERENCE
            score_adjustment += coherence_adjustment
            weight_sum += abs(coherence_adjustment)

        # Calcul final
        final_score = 0.5 + score_adjustment / weight_sum if weight_sum > 0 else
            0.5

        return max(0.0, min(1.0, final_score)) # Assurer que le score reste entre 0
            et 1


    def generate_report(self, input_data, cleaned_text, rule_results, nlp_results,
        external_data, overall_score):
        """
        G n re le rapport final structur , similaire    'RapportEvaluation' du
            PDF.
        """
        report = {
            'idRapport': f"report_{int(datetime.datetime.now().timestamp())}",
            'informationEntree': input_data,
            'dateGeneration': datetime.datetime.now().isoformat(),
            'scoreCredibilite': round(overall_score, 2),
            'resumeAnalyse': "", # Sera g n r  ci-dessous
            'detailsScore': {
                'base': 0.5,
                'adjustments': self._get_score_adjustments(rule_results,
                    nlp_results)
            },
            'sourcesUtilisees': [], # Lister les sources externes consult es
            'reglesAppliquees': rule_results,
            'analyseNLP': { # Filtrer pour ne pas inclure les explications
                potentiellement longues ici
                'sentiment': nlp_results.get('sentiment'),
                'bias_analysis': nlp_results.get('bias_analysis'),
                'named_entities_count': len(nlp_results.get('named_entities', []))
                    ,
                'coherence_score': nlp_results.get('coherence_score'),
                'sentiment_explanation_preview': nlp_results.get('
                    sentiment_explanation', [])[:2] # Juste un aper u
            },
        }

        # G n rer un r sum  textuel simple
        summary_parts = []
        if overall_score > 0.75:
            summary_parts.append("L'analyse sugg re une cr dibilit  LEVE .")
        elif overall_score > 0.55:
            summary_parts.append("L'analyse sugg re une cr dibilit  MOYENNE  
                LEVE .")
        elif overall_score > 0.45:
            summary_parts.append("L'analyse sugg re une cr dibilit  MOYENNE.")
        elif overall_score > 0.25:
            summary_parts.append("L'analyse sugg re une cr dibilit  FAIBLE  
                MOYENNE.")
        else:
            summary_parts.append("L'analyse sugg re une cr dibilit  FAIBLE.")


        if rule_results['source_analysis']['reputation'] != 'Unknown':
            summary_parts.append(f"R putation source : {rule_results['
                source_analysis']['reputation']}.")
        elif isinstance(input_data, str) and is_url(input_data): # Seulement si c'
            tait  une URL mais r putation inconnue
            summary_parts.append("R putation source : Inconnue.")


        if rule_results['linguistic_markers']['sensationalism'] > 0:
            summary_parts.append(f"Marqueurs sensationnalistes d tect s ({
                rule_results['linguistic_markers']['sensationalism']}).")
        if rule_results['linguistic_markers']['doubt'] > 0:
            summary_parts.append(f"Marqueurs de doute d tect s ({rule_results['
                linguistic_markers']['doubt']}).")

        bias_info = nlp_results.get('bias_analysis')
        if bias_info and 'Flagged' in bias_info.get('label',''):
            bias_score_str = f"{bias_info.get('score','N/A'):.2f}" if isinstance(
                bias_info.get('score'), float) else 'N/A'
            summary_parts.append(f"Biais potentiel signal  (Score simul : {
                bias_score_str}).")

        if rule_results['fact_checking']:
            fc_summary = ", ".join([f"{fc['claim']} ({fc['rating']})" for fc in
                rule_results['fact_checking']])
            summary_parts.append(f"V rifications externes trouv es : {fc_summary
                }.")
        elif isinstance(input_data, str) and is_url(input_data): # Si c' tait  une
             URL mais pas de fact check trouv
            summary_parts.append("Aucune v rification externe sp cifique 
                trouv e (simulation).")


        report['resumeAnalyse'] = " ".join(summary_parts)

        # Lister les sources externes (exemple)
        is_input_url_flag = isinstance(input_data, str) and is_url(input_data)
        if is_input_url_flag:
            report['sourcesUtilisees'].append({'type': 'Primary Input URL', 'url':
                input_data})
        report['sourcesUtilisees'].append({'type': 'External Reputation Check', '
            details': f"Reputation: {rule_results['source_analysis']['reputation
            ']} (Simulated)"})
        report['sourcesUtilisees'].append({'type': 'External Fact Check API', '
            details': f"{len(rule_results['fact_checking'])} checks found (
            Simulated)"})


        return report

    def _get_score_adjustments(self, rule_results, nlp_results):
        """Helper pour lister les ajustements de score pour le rapport."""
        adjustments = []
        # Miroir de la logique dans calculate_overall_score, mais juste pour le
            reporting
        if rule_results['source_analysis']['reputation'] == 'High':
            adjustments.append({'factor': 'Source Reputation', 'value': '+High'})
        elif rule_results['source_analysis']['reputation'] == 'Low':
            adjustments.append({'factor': 'Source Reputation', 'value': '-Low'})

        domain_age = rule_results['source_analysis'].get('domain_age_days')
        if domain_age is not None:
            if domain_age > 365 * 2:
                adjustments.append({'factor': 'Domain Age', 'value': '+Old'})
            elif domain_age < 90:
                adjustments.append({'factor': 'Domain Age', 'value': '-New'})

        certainty = rule_results['linguistic_markers']['certainty']
        doubt = rule_results['linguistic_markers']['doubt']
        if certainty > 0 and doubt == 0:
            adjustments.append({'factor': 'Certainty Markers', 'value': f"+{
                certainty}"})
        elif doubt > 0:
            adjustments.append({'factor': 'Doubt Markers', 'value': f"-{doubt}"})

        sensationalism = rule_results['linguistic_markers']['sensationalism']
        if sensationalism > 0:
            adjustments.append({'factor': 'Sensationalism', 'value': f"-{
                sensationalism}"})

        sentiment_info = nlp_results.get('sentiment')
```

```
        if sentiment_info:
            if sentiment_info['label'] == 'NEGATIVE' and sentiment_info['score'] > 0.85:
                adjustments.append({'factor': 'Strong Negative Sentiment', 'value': '-Impact'})

        bias_info = nlp_results.get('bias_analysis')
        if bias_info and 'Flagged' in bias_info.get('label',''):
            bias_score_str = f"{bias_info.get('score', 0):.2f}" if isinstance(bias_info.get('score'), float) else 'N/A'
            adjustments.append({'factor': 'Potential Bias', 'value': f"-Impact (Score: {bias_score_str})"})

        coherence_score = nlp_results.get('coherence_score')
        if coherence_score is not None:
            adj_val = round((coherence_score - 0.5), 2)
            adjustments.append({'factor': 'Coherence (Simulated)', 'value': f"{'+' if adj_val >= 0 else ''}{adj_val}"})

        return adjustments


    def verify_information(self, input_data):
        """
        Pipeline principal pour v rifier la cr dibilit  (adapt  du PDF).
        """
        # V rification initiale du type d'entr e
        if not isinstance(input_data, str) or not input_data.strip():
            print("Erreur: L'entr e fournie n'est pas une cha ne de caract res valide ou est vide.")
            return {"error": "L'entr e doit tre une cha ne de caract res non vide (texte ou URL)."}

        print(f"\n--- V rification pour : {input_data[:100]}... ---")

        # 1. D terminer le type d'entr e et r cup rer le contenu si URL
        text_to_analyze = ""
        is_input_url = is_url(input_data)
        if is_input_url:
            try:
                # --- Simulation ---
                text_to_analyze = fetch_web_content(input_data)
                if text_to_analyze is None: # G rer l' chec  simul  de fetch_web_content
                    print(f" chec de la r cup ration du contenu pour l'URL : {input_data}")
                    return {"error": f"Impossible de r cup rer le contenu de l'URL : {input_data}"}

            except requests.exceptions.RequestException as e: # Garder pour une future impl mentation r elle
                print(f"Erreur r seau/HTTP lors de la r cup ration de l'URL {input_data} : {e}")
                return {"error": f"Erreur r seau lors de la r cup ration de l'URL : {e}"}
            except Exception as e:
                print(f"Erreur inattendue lors du traitement de l'URL {input_data} : {e}")
                return {"error": f"Erreur inattendue lors du traitement de l'URL : {e}"}

        else: # Si ce n'est pas une URL, c'est du texte direct
            text_to_analyze = input_data

        # 2. Pr traitement du texte
        cleaned_text = self.preprocess(text_to_analyze)
        if not cleaned_text:
            # G rer le cas o  le texte est vide apr s nettoyage
            print("Erreur: Le texte est vide apr s pr traitement.")
            return {"error": "Le texte fourni est vide ou ne contient que des lments  supprim s lors du pr traitement."}
        print(f"Texte nettoy  (extrait) : {cleaned_text[:200]}...")

        # 3. R cup rer les donn es externes
        external_data = fetch_external_data(input_data if is_input_url else cleaned_text)
        print(f"Donn es externes (simul es) : {external_data}")

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

        # 5. Analyse bas e sur NLP/IA
        nlp_results = self.nlp_analysis(cleaned_text)
        print(f"R sultats NLP (Sentiment) : {nlp_results.get('sentiment')}")
        print(f"R sultats NLP (Bias) : {nlp_results.get('bias_analysis')}")
        print(f"R sultats NLP (NER count) : {len(nlp_results.get('named_entities', []))}")

        # 6. Calculer le score global
        overall_score = self.calculate_overall_score(rule_results, nlp_results)
        print(f"Score global calcul  : {overall_score:.2f}")

        # 7. G n rer le rapport final
        final_report = self.generate_report(input_data, cleaned_text, rule_results, nlp_results, external_data, overall_score)

        return final_report

# --- Tests du Syst me ---
if __name__ == "__main__":
    system = CredibilityVerificationSystem()
    results = {} # Dictionnaire pour stocker les r sultats

    test_cases = {
        "Test 1 (Texte Simple)": "This post is verified and credible. Avoid false information.",
        "Test 2 (URL Cr dible)": "http://verified-news.com/article123",
        "Test 3 (URL Hoax)": "http://hoax-site.org/the-truth",
        "Test 4 (Texte Sensationnaliste)": "Shocking news! The secret is revealed! This changes everything! Amazing discovery!",
        "Test 5 (Texte Vide)": " ",
        "Test 6 (URL Inexistante)": "http://nonexistent-domain-for-test.xyz",
        "Test 7 (URL G n rique)": "http://example-generic-site.com/page",
        "Test 8 (Texte avec Doute)": "There are rumors and claims that this might be a hoax.",
    }

    import json

    for test_name, test_input in test_cases.items():
        print(f"\n===== Ex cution: {test_name} =====")
        result = system.verify_information(test_input)
        results[test_name] = result # Stocker le r sultat
        print(f"\n--- Rapport Final {test_name} ---")
        print(json.dumps(result, indent=2, ensure_ascii=False))
        print(f"===== Fin: {test_name} =====")

    # Optionnel: Afficher un r sum  de tous les scores  la fin
    print("\n\n===== R sum  des Scores =====")
    for test_name, result in results.items():
        score = result.get('scoreCredibilite', 'Erreur')
        if score == 'Erreur' and 'error' in result:
            score = f"Erreur ({result['error']})"
        print(f"{test_name}: {score}")
```

10. APPENDICE II (TURTLE CODE TO POPULATE THE ONTOLOGY(SUBJECT-PREDICATE-OBJECT)

```
@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#> .
#
#
# ################################################################
# #
# #    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
#
# 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
# #
# #################################################################
#
#
#
#
#
#
#
# Generated by the OWL API (version 4.5.29.2024-05-13T12:11:03Z) https://github.com
      /owlcs/owlapi

<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 .
```