

Approche hybride LLM-Ontologie pour la création d'un graphe de connaissance d'expertise d'accidents aéronautiques

Yvan Valdes TCHOUTOUO TCHEUNJIO¹, Kadidiatou BERTHE¹, Halguièta NASSA TRAWINA³,
Serge SONFACK SOUNCHIO², Franck Anaël MBIAYA⁴, Fiti ISSA SORO⁵, HO Tuong Vinh¹

¹ Université Nationale du Vietnam Hanoï – École Internationale, Vietnam

² Robert Bosch Sp. z o.o., Pologne

³ Laboratoire de Mathématique et d'Informatique, Université Joseph KI-ZERBO, Burkina Faso

⁴ Univ. Orléans, INSA CVL, LIFO EA 4022, France

⁵ LBA Laboratoire de Biomécanique Appliquée, France

8 mai 2026

Résumé

L'analyse des rapports d'accidents aéronautiques est essentielle pour la sécurité civile, mais leur nature textuelle non structurée limite leur exploitation systématique. Ce travail présente une approche automatisée de création de graphe de connaissances d'expertise d'accidents aéronautiques combinant l'ontologie de domaine BAEO (Base Accident Expertise Ontology) et un grand modèle de langage (LLM). Le schéma ontologique contraint l'extraction de triplets RDF, garantissant la cohérence sémantique. Une validation croisée combinant règles structurelles et vérification factuelle par LLM assure la fiabilité du graphe produit. Les expérimentations sur des rapports du Bureau d'Enquêtes et d'Analyses (BEA) montrent une précision d'extraction de 96 % avec Mistral-large. Le graphe structuré facilite la représentation des chaînes de causalité et la réutilisation des connaissances pour l'aide à l'expertise.

Mots-clés

Graphe de connaissances, Ontologie BAEO, LLM, Sécurité aéronautique, Extraction d'information.

Abstract

The analysis of aeronautical accident reports is essential for civil aviation safety, but their unstructured textual nature limits systematic exploitation. This work presents an automated approach for constructing an aeronautical accident expertise knowledge graph combining the BAEO domain ontology (Base Accident Expertise Ontology) and a large language model (LLM). The ontological schema constrains RDF triple extraction, ensuring semantic consistency. A cross-validation step combining structural rules and LLM-based factual verification ensures graph reliability. Experiments on BEA reports show a 96 % extraction precision with Mistral-large. This structured graph enables causal chain representation and supports knowledge reuse for expert analysis.

Keywords

Knowledge Graph, BAEO Ontology, LLM, Aviation Safety, Information Extraction.

1 Introduction

La sécurité aéronautique repose sur l'analyse méthodique des événements passés. En France, le Bureau d'Enquêtes et d'Analyses (BEA) documente ces événements à travers des rapports d'enquête détaillés, couvrant des facteurs techniques, humains et environnementaux. Diffusés sous forme de PDF non structurés, ces documents limitent leur exploitation systématique et la corrélation à grande échelle. Les Graphes de Connaissances (KG) constituent une solution éprouvée pour intégrer des données hétérogènes et représenter des connaissances complexes [3, 9].

Transformer ce récit narratif en ressource formelle interrogeable pose un double défi : la complexité de l'extraction d'expertise métier et la garantie de conformité aux modèles de données existants. Les LLM offrent des capacités d'extraction prometteuses, mais souffrent d'hallucinations factuelles et d'un manque de connaissances spécialisées [4]. L'ontologie BAEO [8] fournit le cadre conceptuel pour modéliser ces connaissances, mais son peuplement manuel est chronophage.

Nous proposons une approche automatisée guidée par BAEO qui, contrairement aux méthodes non contraintes, injecte classes et propriétés ontologiques directement dans la génération du LLM [2, 6]. L'originalité de ce travail réside dans une boucle de validation croisée combinant contraintes structurelles et vérification de fidélité factuelle. La suite de l'article est organisée comme suit : la section 2 décrit l'ontologie BAEO ; la section 3 présente l'état de l'art ; la section 4 détaille l'architecture ; la section 5 discute les résultats ; la section 6 conclut.

2 Fondement sémantique : L'Ontologie BAEO

Le succès et la pertinence de notre pipeline de génération de graphe reposent sur un fondement sémantique rigoureux. Notre choix s'est porté sur BAEO (Base Accident Expertise Ontology), une ontologie formelle spécifiquement conçue pour la représentation des connaissances issues de l'expertise d'accidents, comme le détaille l'article de recherche de [8]. L'article souligne que les représentations traditionnelles, telles que les documents textuels ou les bases de données relationnelles, sont insuffisantes pour capturer la complexité et la sémantique des liens de causalité inhérents à un accident. BAEO a été développée pour combler cette lacune en proposant un modèle de connaissance formel, réutilisable et interopérable, illustré par la Figure 1.

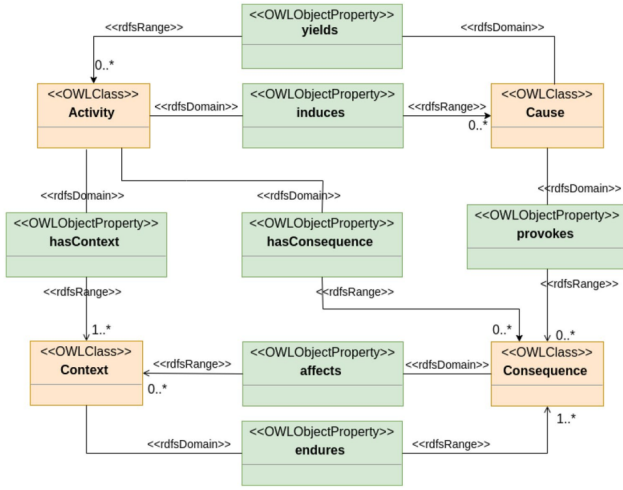


FIGURE 1 – Modèle conceptuel de l'ontologie BAEO, illustrant les relations principales entre les concepts fondamentaux de l'expertise d'accident [8]

Notre décision de nous appuyer sur BAEO est motivée par plusieurs de ses caractéristiques fondamentales :

1. **Spécificité et généralité** : BAEO est une ontologie de domaine (couche intermédiaire de la pyramide ontologique), ce qui la rend plus spécifique qu'une ontologie de haut niveau (comme DOLCE ou BFO), mais suffisamment générale pour être étendue à divers types d'accidents (aéronautique, routier, industriel). Cette modularité, illustrée dans l'article, est un atout majeur pour la pérennité de notre approche.
2. **Richesse conceptuelle** : L'ontologie est structurée autour de cinq concepts fondamentaux indispensables à l'expertise d'accident :
 - **Activity** : actions ou processus (remise des gaz, interruption d'approche);
 - **Context** : environnement – *Agent* (Pilote, Avion), *Location*, *Environment*;
 - **Cause** : facteurs déclencheurs (*HumanCause*, *SystemCause*, *NaturalCause*);

- **Consequence** : résultats (*Fatality*, *Failure*, *Destruction*);
- **Time** : modélisation temporelle via OWL-Time (13 relations d'Allen).

3. **Conception rigoureuse et validée** : BAEO n'est pas une simple taxonomie; elle a été conçue en suivant une méthodologie rigoureuse de type MDA (Model- Driven Architecture), utilisant UML pour la modélisation conceptuelle avant d'être implémentée avec les technologies du web sémantique (OWL 2 DL). Elle intègre également des mécanismes de raisonnement, comme les chaînes de propriétés, pour inférer de nouvelles connaissances (par exemple, lier un agent à une conséquence via une chaîne de causalité).

D'un point de vue opérationnel, BAEO est exploitée opérationnellement : ses classes définissent les types d'entités autorisés, ses propriétés (avec leurs `rdfs:domain/rdfs:range`) spécifient les relations possibles. Ces éléments contraignent directement la génération des triplets par le LLM et valident leur cohérence sémantique.

3 Travaux antérieurs

La construction de graphes de connaissances (KG) à partir de sources textuelles est un domaine de recherche dynamique. Comme le définissent [3], les KG permettent d'intégrer des données hétérogènes dans un cadre formel facilitant le raisonnement. Ce domaine est marqué par le passage des méthodes classiques basées sur la reconnaissance d'entités nommées (NER) enrichie par les graphes [7] vers des approches sémantiques fondées sur les grands modèles de langage (LLM).

3.1 Construction de KG assistée par LLM

L'utilisation des LLM pour traiter des documents hétérogènes a ouvert de nouvelles perspectives. Le framework Docs2KG [9] extrait des informations multimodales à partir de divers formats (PDF, Excel) en un graphe structuré. [4] montrent que l'association LLM-KG pallie les hallucinations des modèles seuls en combinant compréhension linguistique et rigueur factuelle. [6] démontrent que l'usage d'une ontologie de domaine est crucial pour guider l'extraction dans des corpus spécialisés. Cette thématique Text2KG fait l'objet de workshops dédiés (ISWC, ESWC) et de benchmarks d'évaluation comme WebNLG et KELM, qui permettent de situer les systèmes d'extraction par rapport à l'état de l'art.

3.2 Approches guidées par l'ontologie (Ontology-grounded)

Pour garantir la précision sémantique, plusieurs travaux couplent schémas formels et LLM. [2] présentent une méthode où la génération est contrainte par une ontologie de base (ex. Wikidata). Contrairement aux systèmes bout-en-bout comme Grapher [5], qui s'appuient sur des têtes d'extraction entraînées, les approches guidées par l'ontolo-

gie utilisent le schéma comme garde-fou sémantique. [10] montrent que ces pipelines surpassent les modèles généralistes (GraphGPT) pour des domaines pointus tels que l'électronique automobile. [1] introduisent OntoKGen, un pipeline interactif exploitant le Chain-of-Thought (CoT) pour aligner la génération sur les préférences utilisateur.

3.3 Expertise d'accidents aéronautiques

Dans le domaine aéronautique, les LLM ont été explorés pour l'assistance à la rédaction de rapports d'enquête (NTSB, BEA), illustrant un intérêt croissant pour l'automatisation de tâches documentaires à haute criticité. Sur le plan sémantique, les travaux de [8] ont posé les bases avec l'ontologie BAEO (*Base Accident Expertise Ontology*). Notre travail s'inscrit dans la continuité de ces recherches en automatisant le peuplement de BAEO via une approche hybride LLM-ontologie, garantissant la cohérence structurale des triplets extraits tout en exploitant la capacité de compréhension contextuelle de Mistral-large.

4 Solution proposée

4.1 Architecture du pipeline

Le pipeline transforme des rapports PDF du BEA en un graphe RDF en trois phases s'appuyant directement sur BAEO (Figure 3).

Phase 1 – Prétraitement : Les rapports sont segmentés sémantiquement via spaCy (fr_core_news_lg), en trois étapes : extraction PDF (fast, langue fra), nettoyage et normalisation, puis segmentation par phrases. Le modèle *large* gère les abréviations aéronautiques (ft, kt, ILS) et la ponctuation technique des rapports BEA. Un contexte glissant de 3 chunks (précédent + principal + suivant) permet la résolution de coréférences.

Phase 2 – Extraction guidée par LLM : Un *Ontology Handler* exécute deux requêtes SPARQL internes sur `baeo_application.ttl` : la première extrait toutes les `owl:Class` avec leurs `rdfs:label` ; la seconde extrait toutes les `owl:ObjectProperty` et `owl:DatatypeProperty` avec `domain/portée`. Les listes résultantes (`class_names`, `property_names`) sont injectées dans le prompt du LLM :

```
### ONTOLOGIE DISPONIBLE :
- Classes : {'','.join(class_names)}
- Propriétés : {'','.join(property_names)}
```

Phase 3 – Validation et insertion : C'est une phase cruciale pour garantir la qualité. Les triplets bruts passent par deux étapes complémentaires :

(1) un *OntologicalValidator* vérifie la compatibilité des types sujet/objet avec les axiomes `rdfs:domain/rdfs:range` ; (2) le LLM vérifie la fidélité factuelle via un prompt binaire (VRAI/FAUX). Seuls les triplets validés par les deux étapes sont insérés dans le graphe final via RDFLib.

4.2 Guidage du LLM par l'ontologie

Le Tableau 1 détaille la structure du prompt d'extraction. Le processus Chain-of-Thought impose : (1) identification

des entités ; (2) validation ontologique mentale de chaque relation ; (3) vérification factuelle stricte ; (4) enrichissement par `human_readable_label` et `comment` ; (5) production du JSON uniquement pour les éléments validés. Cette contrainte réduit les hallucinations structurales en limitant l'espace de génération au vocabulaire BAEO.

TABLE 1 – Structure du prompt d'extraction guidée par l'ontologie

INSTRUCTION : Tu es un expert en extraction sémantique. Analyse le 'TEXTE PRINCIPAL' en utilisant le contexte et extrais les entités et relations dans un format JSON strict.
PROCESSUS DE RÉFLEXION (CHAIN OF THOUGHT) : 1. Identification : Identifie toutes les entités et relations potentielles. 2. Validation Ontologique : Pour chaque relation, vérifie mentalement qu'elle respecte le schéma de l'ontologie. 3. Validation Factuelle : Assure-toi que chaque fait est EXPLICITEMENT mentionné dans le texte principal. 4. Enrichissement : Génère un 'human_readable_label', 'comment', et 'description'. 5. Formatage Final : Ne produis le JSON qu'avec les entités et relations qui ont passé toutes tes validations internes.
RÈGLES STRICTES : - Les valeurs pour 'type' et 'predicate' doivent IMPÉRATIVEMENT provenir de l'ontologie DISPONIBLE. - Assigne UN SEUL 'type' par entité (le plus précis). - Résous les coréférences (pronoms) en utilisant le contexte.
ONTOLOGIE DISPONIBLE : - Classes : [LISTE_DES_CLASSES_BAEO] - Propriétés : [LISTE_DES_PROPRIETES_BAEO]
FORMAT ATTENDU : {"entities": [{"label": "Accident_HB-LSD", "human_readable_label": "Accident du Piper PA34", "type": "Event", "properties": [{"predicate": "hasLocation", "object": "Aerodrome_Bale"}]}}

4.3 Limites du validateur

Le validateur filtre les triplets dont le type sujet/objet contredit les axiomes `rdfs:domain/rdfs:range` de l'ontologie. Les triplets avec des prédicats inconnus passent la validation sans être bloqués. Les causes d'échec identifiées comprennent : entités orphelines (jamais référencées), omission de *hasLocation* pour des événements, et confusion de types (TRS vs NonPhysicalResource).

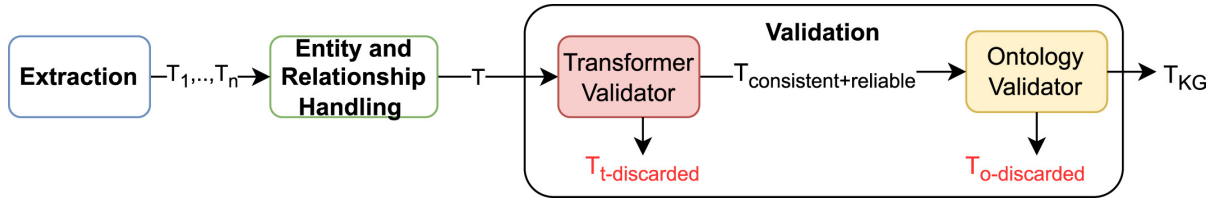


FIGURE 2 – Processus de validation automatique des triplets

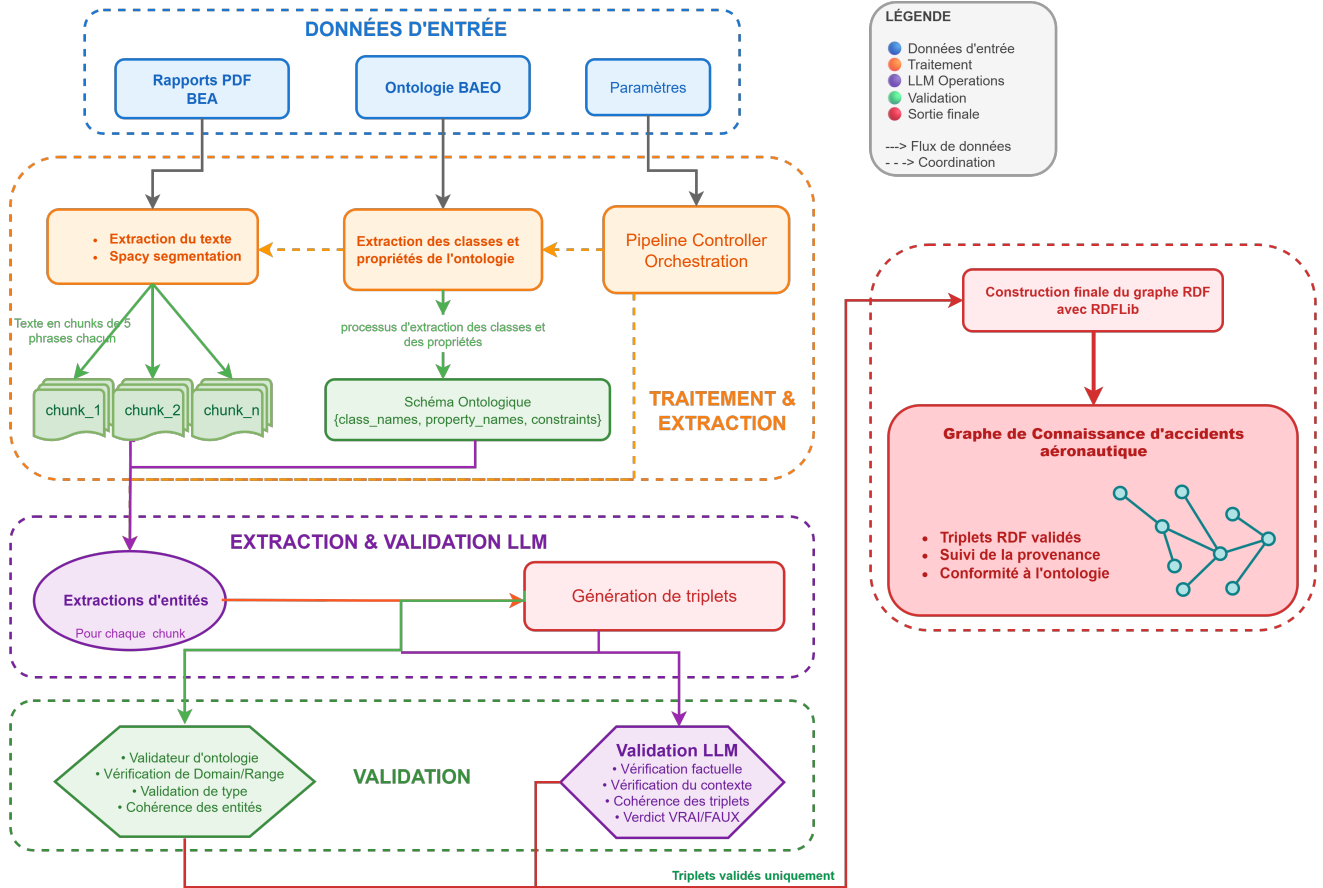


FIGURE 3 – Architecture du pipeline de génération du KG d'expertise aéronautique.

4.4 Expérimentation

4.4.1 Corpus et modèles évalués

Trois LLMs à poids ouverts, déployables localement, ont été évalués : Mistral-7B-Instruct-v0.3, Llama-3.1-8B-Instruct (contrainte GPU et API gratuite HuggingFace), et Mistral-large (API mistral-large-latest). Le corpus comprend trois rapports PDF du BEA : BEA2016-0759 (*Piper PA34-200T HB-LSD*, Bâle-Mulhouse, 07/12/2016), segmenté en ~160 chunks, et deux rapports complémentaires (9H-MDJ, 21ANN). L'ontologie (baeo_application.ttl) et les 20 requêtes SPARQL d'évaluation sont disponibles en accès libre sur le dépôt du projet : <https://github.com/Tchoutouo/KG-Builder>.

4.4.2 Protocole d'évaluation

L'évaluation repose sur 100 triplets annotés manuellement par modèle, puis sur 20 requêtes SPARQL de référence couvrant cinq catégories : chronologie des événements (Q1–Q4), facteurs humains/pilote (Q5–Q8), facteurs techniques/aéronef (Q9–Q12), facteurs environnementaux (Q13–Q15), analyse et conclusions (Q16–Q20). Les métriques calculées sont la précision, le rappel et le F1-Score.

4.4.3 Résultats

Le Tableau 2 synthétise les performances comparatives. Mistral-large se distingue nettement par sa capacité à suivre les contraintes ontologiques complexes.

Mistral-large a produit des triplets précis et un graphe propre. Les 20 requêtes SPARQL ont toutes retourné des résultats corrects et pertinents. En revanche, Llama-

cause	label	comment
baeo:Conditions_Lvp_Hb_Lsd	"Conditions de faible visibilité (LVP) lors de l'approche du HB-LSD"	"Conditions météorologiques caractérisées par une faible visibilité (Low Visibility Procedures, LVP), réduisant la capacité du pilote à distinguer clairement les repères visuels fiables. Ce contexte a probablement amplifié le risque de désorientation spatiale et d'erreur d'appréciation."
baeo:Conditions_Lvp_Hb_Lsd	"Conditions de faible visibilité (LVP) lors de l'approche du HB-LSD"	"Conditions météorologiques réduisant drastiquement la visibilité extérieure."
baeo:Conditions_Lvp_Hb_Lsd	"Conditions de faible visibilité (LVP) lors de l'approche du HB-LSD"	"Contexte météorologique défavorable ayant pu aggraver les difficultés de perception."
baeo:Contexte_Vol_Nuit_Basse_Visibilite	"Contexte de vol de nuit en basse visibilité"	"Environnement opérationnel défavorable caractérisé par l'absence de repères visuels."
baeo:Contexte_Vol_Nuit_Basse_Visibilite	"Contexte de vol de nuit en basse visibilité"	"Situation météorologique et temporelle (nuit) privant le pilote de références extérieures (horizon, sol, autres aéronefs). Ce contexte a amplifié les risques de désorientation spatiale, surtout lors de manœuvres complexes (virages, montée rapide). Les conditions LVP (Low Visibility Procedures) étaient probablement en vigueur, limitant encore les aides visuelles."

FIGURE 8 – Requête analyse : causes identifiées de la désorientation spatiale.

extraites. Les 4 % d'écart concernent principalement des relations implicites et des nuances contextuelles.

Dans notre pipeline, la validation couvre les contraintes structurelles de base (`rdfs:domain/rdfs:range`). L'intégration de règles SHACL constitue une perspective naturelle pour des contraintes de cardinalité plus expressives, potentiellement couplée à une approche "LLM juge" [10].

Plusieurs limites subsistent : (1) la présence de tableaux et schémas techniques dans les PDF peut rompre la continuité sémantique des chunks (des travaux récents sur le parsing PDF structuré adressent ce point); (2) la sensibilité au modèle est marquée – la chute de Llama-3.1-8B souligne le besoin de modèles à grande fenêtre de contexte; (3) la version 70B de Llama fournirait des résultats probablement très différents, contrainte par les ressources GPU disponibles. Enfin, un retour d'expérience d'enquêteurs du BEA renforcerait la crédibilité pratique du système.

6 Conclusion

Nous avons présenté un pipeline automatisé transformant des rapports d'accidents aéronautiques non structurés en un graphe RDF d'expertise, avec une précision de 96 % pour Mistral-large. L'extraction guidée par l'ontologie BAEO réduit les biais des LLM tout en préservant la richesse de l'expertise métier.

Plusieurs perspectives se dégagent : (1) **Multimodalité** – intégrer images, diagrammes et trajectoires de vol pour enrichir le graphe; (2) **Agent conversationnel** – permettre aux experts d'interroger la base en langage naturel via un

DMS robuste [10]; (3) **Validation humaine en boucle** – interface de validation manuelle des triplets par des enquêteurs du BEA; (4) **SHACL + LLM juge** – renforcement de la validation déclarative; (5) **Fine-tuning** – adapter un LLM léger sur des rapports BEA annotés pour réduire les coûts computationnels.

Références

- [1] Mohammad Sadeq Abolhasani and Rong Pan. Leveraging llm for automated ontology extraction and knowledge graph generation. *arXiv preprint arXiv :2412.00608*, 2024.
- [2] Xiaohan Feng, Xixin Wu, and Helen Meng. Ontology-grounded automatic knowledge graph construction by llm under wikidata schema. *arXiv preprint arXiv :2412.20942*, 2024.
- [3] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard De Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, et al. Knowledge graphs. *ACM Computing Surveys (Csur)*, 54(4) :1–37, 2021.
- [4] Chuangtao Ma, Yongrui Chen, Tianxing Wu, Arijit Khan, and Haofen Wang. Unifying large language models and knowledge graphs for question answering : Recent advances and opportunities. In *EDBT*, pages 1174–1177, 2025.
- [5] Igor Melnyk, Pierre Dognin, and Payel Das. Knowledge graph generation from text. In *Findings of the Association for Computational Linguistics : EMNLP 2022*, pages 1610–1622, 2022.
- [6] Zhixin Meng, Shaoxiong Zhan, Ruiqing Xu, Wolfgang Mayer, Ye Zhu, Hong-Yu Zhang, Chuan He, Keqing He, Debo Cheng, and Zaiwen Feng. Domain ontology-driven knowledge graph generation from text. *ACM Transactions on Probabilistic Machine Learning*, 1(4) :1–18, 2025.
- [7] David Muñoz, Fernando Pérez, and David Pinto. Named entity recognition based on a graph structure. *Computación y Sistemas*, 24(2) :553–563, 2020.
- [8] Serge Sonfack Souchio, Bernard Kamsu-Foguem, and Laurent Geneste. Construction of a base ontology to represent accident expertise knowledge. *Cognition, Technology & Work*, 25(2) :183–201, 2023.
- [9] Qiang Sun, Yuanyi Luo, Wenxiao Zhang, Sirui Li, Jichunyang Li, Kai Niu, Xiangrui Kong, and Wei Liu. Docs2kg : Unified knowledge graph construction from heterogeneous documents assisted by large language models. *arXiv preprint arXiv :2406.02962*, 2024.
- [10] Frank Wawrzik, Matthias Plaue, Savan Vekariya, and Christoph Grimm. Customized information and domain-centric knowledge graph construction with large language models. *arXiv preprint arXiv :2409.20010*, 2024.