Vers une synthèse neuro-symbolique des dossiers médicaux électroniques : de GraphSynth aux modèles de langage multimodaux

Amal Beldi*, Salma Sassi**
Richard Chbeir***, Abderrazek Jemai****

* Laboratoire LIPN, Université Sorbonne Paris Nord, France
beldi@lipn.univ-paris13.fr ** Central School of Electronics (ECE), Lyon, France
ssassi@ece.fr *** LIUPPA, Université de Pau et des Pays de l'Adour, France
richard.chbeir@univ-pau.fr **** SERCOM Laboratory, University of Tunis El Manar, Tunisia
abderrazekjemai@yahoo.co.uk

Résumé. Les grands modèles de langage multimodaux (MLLMs) ouvrent des perspectives importantes pour la synthèse et l'analyse des dossiers médicaux électroniques (DME), mais restent fortement limités par les hallucinations, le manque de traçabilité du raisonnement et une faible intégration des connaissances structurées issues de graphes ou d'ontologies.

Dans cet article, nous proposons une architecture neuro-symbolique pour la synthèse de DME fondée sur deux briques principales développées dans une thèse : (i) *GraphSynth*, qui modélise les données hétérogènes du dossier sous la forme d'un graphe de données et les résume à l'aide d'une famille d'opérateurs dédiés ; (ii) *RDF-GraphSyn*, qui traduit ces graphes résumés en graphes de connaissances RDF optimisés.

Ces graphes constituent ensuite une couche symbolique permettant de conditionner un MLLM dans un schéma de *Retrieval-Augmented Generation* (RAG) et de contrôler ses sorties par une vérification guidée par le graphe. Nous montrons comment ce cadre permet de produire des réponses plus factuelles et explicables, chacune étant associée à un sous-graphe de preuves. Un cas d'usage portant sur la synthèse d'un dossier de patient atteint de diabète de type 2 illustre l'approche proposée.

Mots-clés : Dossiers médicaux électroniques, graphes de connaissances, neuro-symbolique, grands modèles de langage, RAG, explicabilité.

1 Introduction

Les dossiers médicaux électroniques (DME) constituent une source d'information essentielle pour le suivi des patients. Toutefois, leur volume, leur hétérogénéité (notes

cliniques, examens, mesures, imagerie) et leur évolution temporelle rendent leur exploitation difficile, en particulier dans des contextes cliniques contraints tels que les consultations rapides ou les situations d'urgence. Les grands modèles de langage (LLMs), et plus récemment leurs versions multimodales (MLLMs), offrent des perspectives prometteuses pour assister les cliniciens, notamment pour la génération de résumés cliniques ou la réponse à des questions sur l'historique du patient. Cependant, leur utilisation directe reste limitée par plusieurs problèmes majeurs : hallucinations factuelles, manque de traçabilité du raisonnement et faible intégration des connaissances structurées.

Les approches symboliques, en particulier les graphes de connaissances, offrent un cadre pertinent pour structurer les données hétérogènes des DME et expliciter les relations entre entités cliniques. Dans ce contexte, une thèse récente a proposé deux briques complémentaires : (i) *GraphSynth*, qui modélise les données du DME sous forme de graphe de données et permet leur synthèse à l'aide d'opérateurs spécifiques (display, filtrate, transformate, calculate, abstract) ; (ii) *RDF-GraphSyn*, qui transforme ces graphes résumés en graphes de connaissances RDF optimisés grâce à l'algorithme *HERSE*.

Dans cet article, nous montrons comment ces briques constituent le socle d'une *architecture neuro-symbolique* pour la synthèse des DME à l'aide de MLLMs. L'idée centrale est de construire un *graphe de connaissances patient-centrique* à partir du DME, puis de l'utiliser comme mémoire externe dans un cadre de *Retrieval-Augmented Generation* (RAG). Le graphe guide ainsi la sélection du contexte, contraint la génération du modèle et permet de vérifier les réponses produites à l'aide d'une *vérification grapho-guidée*, chaque réponse étant associée à un sous-graphe de preuves.

Nous illustrons cette approche sur un cas d'usage portant sur la synthèse du dossier d'un patient atteint de diabète de type 2, incluant la génération de résumés cliniques, la réponse à des questions ciblées et l'adaptation du contenu au profil du médecin via *UserProfile-Graph*. Les contributions principales de cet article sont les suivantes :

— une architecture neuro-symbolique combinant GraphSynth, RDF-GraphSyn et un MLLM dans un pipeline complet de synthèse ;
— un algorithme global décrivant les différentes étapes du pipeline ;
— un module de vérification grapho-guidée produisant un score de cohérence et un sous-graphe explicatif ;
— un cas d'usage sur le diabète de type 2 illustrant la réduction des hallucinations et l'amélioration de l'explicabilité.

Le reste de l'article est organisé comme suit : la Section 2 présente les travaux connexes ; la Section 3 décrit les briques symboliques ; la Section 4 présente l'architecture proposée ; la Section 5 détaille le protocole expérimental ; et la Section 6 discute les apports et limites.

2 Contexte et travaux connexes

Notre proposition se situe à l'intersection de quatre axes de recherche : (i) la synthèse de dossiers médicaux électroniques (DME), (ii) les grands modèles de langage (LLMs / MLLMs) et la génération augmentée par recherche (RAG), (iii) les graphes de connaissances et leur résumé, (iv) les approches neuro-symboliques.

2.1 Synthèse de DME et résumé clinique

Les travaux sur le résumé clinique ont évolué des modèles séquence-à-séquence vers les architectures Transformer et les LLMs pré-entraînés Devlin et al. (2019). Plusieurs modèles spécialisés (ClinicalBERT, BioGPT, Med-PaLM) ont montré leur capacité à produire des résumés exploitables pour les notes et séjours hospitaliersAlsentzer et al. (2019) Gu et al. (2023)Singhal et al. (2023) Cependant, ces approches restent centrées sur le texte et exploitent peu la structure temporelle et relationnelle du DME Beldi et al. (2022b). De plus, la vérification des hallucinations factuelles reste un défi important.

2.2 LLMs, hallucinations et RAG

Les LLMs souffrent de productions plausibles mais incorrectes Huang et al. (2025)Singhal et al. (2023). Les méthodes RAG Lewis et al. (2020) atténuent partiellement ce problème en injectant des documents pertinents dans le contexte. Plusieurs travaux appliquent RAG aux DME ou aux guidelines et montrent un gain de factualité, mais la traçabilité et le contrôle des réponses demeurent limités Culié (2024). Notre approche substitue au RAG textuel un *RAG symbolique* fondé sur un graphe de connaissances centré patient.

2.3 Graphes de connaissances et graph summarization

Les graphes biomédicaux (UMLS, SNOMED CT, ICD) sont largement utilisés pour structurer la connaissance Schulz et al. (2009),Hogan et al. (2021). Leur taille et leur complexité ont conduit au développement de méthodes de *graph summarization*, permettant d'obtenir des sous-graphes compacts et orientés tâche Liu et al. (2018). Dans le contexte RDF, plusieurs approches proposent des schémas ou résumés destinés à la visualisation ou à l'optimisation de requêtes Kondylakis et al. (2019). Notre framework RDF-GraphSyn s'inscrit dans cette lignée en générant des graphes RDF résumés et optimisés à partir d'un graphe DME construit avec GraphSynth.

2.4 LLMs augmentés par graphes et neuro-symbolique

L'intégration de graphes dans les LLMs suscite un intérêt croissant Agrawal et al. (2024). Les stratégies vont de l'injection de faits dans le prompt à l'usage de graphes comme mémoire externe ou à la génération guidée par SPARQL. Par ailleurs, les architectures neuro-symboliques combinent composants neuronaux et structures logiques ou graphiques pour améliorer robustesse et explicabilité Schulz et al. (2009).

Pour analyser les approches existantes dans le contexte DME, nous retenons quatre critères : (C1) factualité, (C2) traçabilité, (C3) exploitation de la structure du DME, (C4) personnalisation au profil médecin. Le tableau 1 met en évidence que notre cadre neuro-symbolique (GraphSynth + RDF-GraphSyn + MLLM + vérification graphe-guidée) répond conjointement à ces exigences.

Approche	**Factualité / traçabilité** (C1–C2)	**Structure exploitée** (C3)	**Limites principales** (vs C1–C4)
LLMs sans contexte	Hallucinations fréquentes ; aucune traçabilité vers le DME.	Aucune structure (texte brut linéaire).	Manque de contrôle, pas de preuves, aucune personnalisation.
RAG textuel classique	Factualité moyenne, dépendante du retrieval ; traçabilité limitée aux passages sources.	Segments textuels isolés (notes, rapports, guidelines).	Contexte volumineux et bruité ; vue fragmentée du DME.
Graphes de connaissances externes (UMLS, SNOMED)	Factualité élevée pour la connaissance biomédicale générale.	Ontologies / KGs biomédicaux génériques.	Non centrés patient ; pas d'intégration temporelle ni clinique fine.
Knowledge Graph Summarization	Bonne factualité si le graphe source est robuste.	Sous-graphes compressés ou schémas simplifiés.	Perte d'information ; résumés peu orientés tâche ou utilisateur.
Neuro-symbolique générique	Traçabilité bonne si les règles sont explicites.	Graphes + règles logiques + programmes symboliques.	Implémentation complexe ; rare en contexte DME réel.
Notre approche	Factualité élevée via vérification grapho-guidée et contraintes RDF.	Graphe résumé patient-centré (GraphSynth + RDF-GraphSyn).	Dépend de la qualité du graphe ; nécessite pipeline dédié mais permet personnalisation (UP) et explications par sous-graphe.

TABLE 1 – *Etude comparative des approches de synthèse de DME.*

3 Briques symboliques pour la synthèse des DME

Cette section présente les briques symboliques constituant la couche centrale de notre architecture : le métamodèle *GraphSynth*, l'algorithme *DGsumm* pour la génération de graphes résumés Beldi (2024), les modules *Topic Graph Summary* et *UserProfile-Graph* pour la personnalisation Beldi et al. (2022a), ainsi que *RDF-GraphSyn* et l'algorithme *HERSE* pour la synthèse RDF Beldi et al. (2024).

3.1 Graphe de données et framework GraphSynth

GraphSynth repose sur un *graphe de données étiqueté* $DG = (V, E, \lambda)$ où les noeuds représentent documents, événements cliniques, mesures et diagnostics, et les arcs codent des relations temporelles ou sémantiques. Le graphe est construit à partir des données hétérogènes du DME en suivant un schéma formel Beldi et al. (2022c). Il sert de *modèle*

pivot pour unifier les données et préparer la synthèse. GraphSynth fournit également des mécanismes de versionnage pour suivre l'évolution du DME.

3.2 Opérateurs de synthèse et algorithme DGsumm

La synthèse repose sur cinq familles d'opérateurs Beldi et al. (2023) : `display` (sélection), `filtrate` (filtrage), `transformate` (agrégation / simplification structurelle), `calcule` (indicateurs cliniques) et `abstract` (concepts et résumés synthétiques). L'algorithme DGSUMM combine ces opérateurs selon la tâche clinique T et le profil utilisateur UP, produisant un graphe résumé compact GS orienté vers l'usage (par ex. trajectoire diabétique, événements récents).

3.3 Personnalisation : Topic Graph Summary et UserProfile-Graph

Pour adapter le résumé au profil du médecin, nous utilisons *UserProfile-Graph* et *Topic Graph Summary* . Des thèmes sont extraits via un modèle de sujets (hLDA), puis organisés en *graphe de sujets* Beldi et al. (2022a). Le profil utilisateur pondère ces thèmes et oriente la sélection des parties pertinentes de GS, permettant un résumé plus ciblé (ex. endocrinologue vs généraliste) tout en contrôlant la taille du contexte transmis au MLLM.

3.4 RDF-GraphSyn et HERSE : génération et optimisation RDF

RDF-GraphSyn traduit GS en un graphe de connaissances RDF G_{RDF} Beldi et al. (2023), en générant classes, propriétés (TBox) et individus (ABox) selon les ontologies cliniques. L'algorithme *HERSE* optimise ce graphe en détectant et en réécrivant les noeuds anonymes, produisant un graphe plus compact et plus interrogeable G^*_{RDF}. Dans notre architecture, ce graphe constitue la mémoire symbolique utilisée pour : (i) construire le contexte RAG, (ii) vérifier les réponses du MLLM, (iii) générer un sous-graphe de preuves.

4 Architecture neuro-symbolique proposée

Sur la base des briques symboliques décrites dans la Section 3, nous proposons une architecture neuro-symbolique qui combine : (i) un pipeline de synthèse et de structuration des données du DME par graphes (GraphSynth, DGsumm, Topic Graph Summary, UserProfile-Graph, RDF-GraphSyn, HERSE) ; (ii) un module de génération en langage naturel basé sur un LLM ou MLLM ; (iii) un module de vérification graphe-guidée et d'explication.

L'architecture peut être décrite comme une chaîne de traitement en plusieurs étapes (Figure 1)

1. **Ingestion et intégration du DME** : les données brutes du dossier (documents textuels, comptes rendus, résultats d'examens, mesures, etc.) sont extraites depuis

les systèmes d'information cliniques et normalisées (formats de dates, unités, identifiants).

2. **Construction du graphe de données DG (GraphSynth)** : les données intégrées sont projetées dans un graphe de données étiqueté DG à l'aide du schéma de graphe défini dans le cadre de *GraphSynth*. Ce graphe relie les entités cliniques (consultations, examens, diagnostics, traitements, mesures, etc.) et leurs métadonnées.
3. **Synthèse orientée tâche et profil (DGsumm, topics, profil)** : l'algorithme DGSUMM applique les opérateurs de synthèse (display, filtrate, transformate, calculate, abstract) pour produire un graphe résumé GS adapté à une tâche clinique T (par exemple, "préparer la consultation", "résumer la trajectoire diabétique") et à un profil utilisateur UP (par exemple, médecin généraliste vs endocrinologue). Topic Graph Summary et UserProfile-Graph interviennent pour orienter la synthèse vers les sujets d'intérêt de l'utilisateur.
4. **Traduction en graphe de connaissances RDF (RDF-GraphSyn, HERSE)** : le graphe résumé GS est traduit en graphe RDF G_{RDF} par *RDF-GraphSyn*, puis optimisé en G^*_{RDF} par l'algorithme *HERSE*, qui traite notamment les nœuds anonymes. Le graphe G^*_{RDF} sert de *base de connaissances* centrée sur le DME du patient.
5. **Construction du contexte symbolique pour le MLLM** : un module BUILDCONTEXT sélectionne, dans G^*_{RDF}, les sous-graphes et triplets pertinents pour la tâche T et le profil UP, puis les linéarise sous une forme textuelle ou semi-structurée. Un module BUILDPROMPT assemble ce contexte avec les instructions et la question clinique pour construire le prompt P.
6. **Génération de la réponse par le MLLM** : le prompt P est soumis à un LLM ou MLLM (par exemple un modèle de type GPT ou un modèle biomédical spécialisé), produisant une réponse textuelle brute A_{raw} qui peut être post-traitée en A (normalisation, simplification).
7. **Vérification grapho-guidée et explication** : un module VERIFYANDEXPLAIN confronte la réponse A au graphe G^*_{RDF}, en extrayant des entités et relations, en vérifiant si elles sont supportées par le graphe, et en construisant un sous-graphe d'explication G_{exp}. Un statut global (`OK` / `PARTIEL` / `KO`) est associé à la réponse.
8. **Formatage de la sortie** : la réponse textuelle A est combinée avec le statut de vérification et le sous-graphe G_{exp} pour produire une sortie destinée au clinicien : texte explicatif et, éventuellement, visualisation graphique des éléments factuels mobilisés.

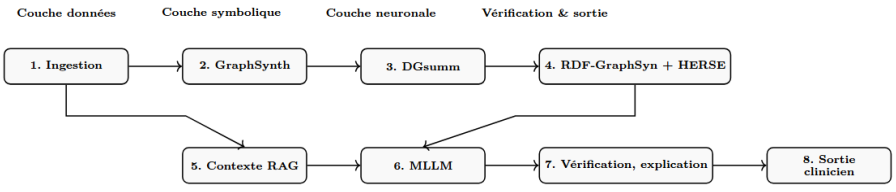

FIGURE 1 – *Architecture neuro-symbolique : couplage entre couche symbolique et couche neuronale (MLLM).*

4.1 Algorithme global de synthèse neuro-symbolique

L'algorithme 1 formalise le pipeline complet de synthèse neuro-symbolique. À partir du DME, il construit un graphe résumé orienté tâche et profil, le traduit en un graphe RDF optimisé, puis en dérive un contexte symbolique pour guider la génération du MLLM. La réponse produite est enfin vérifiée par comparaison au graphe et enrichie d'un sous-graphe de preuves.

Algorithme 1 : Pipeline neuro-symbolique pour la synthèse de dossiers médicaux électroniques

Input : D : données du dossier médical électronique (DME) pour un patient ;
T : tâche clinique (résumé, question-réponse, etc.) ;
UP : profil utilisateur (UserProfile-Graph) ;
$\mathcal{O}$: ontologies / schémas métiers ;
Θ : paramètres de configuration (seuils, règles, etc.).
Output : R : réponse textuelle finale ;
G_{exp} : sous-graphe d'explication associé à R.

Étape 1 : construction du graphe de données
$DG \leftarrow \textsc{BuildDG}(D, \mathcal{O})$ `// Projection du DME dans un graphe de données DG`

Étape 2 : synthèse orientée tâche et utilisateur
$GS \leftarrow \text{DG}\textsc{Summ}(DG, T, UP)$ `// Résumé graphe via display/filtrate/transformate/calculate/abstract`

Étape 3 : passage au graphe de connaissances RDF
$G_{\text{RDF}} \leftarrow \text{RDF}\textsc{GraphSyn}(GS, \mathcal{O})$ `// Traduction DG → RDF`
$G^*_{\text{RDF}} \leftarrow \text{HERSE}(G_{\text{RDF}})$ `// Optimisation, réduction des nœuds anonymes`

Étape 4 : construction du contexte symbolique pour le MLLM
$C \leftarrow \textsc{BuildContext}(G^*_{\text{RDF}}, T, UP)$ `// Sélection / linéarisation de sous-graphes pertinents`
$P \leftarrow \textsc{BuildPrompt}(C, T)$ `// Prompt complet : instructions + contexte + tâche`

Étape 5 : interrogation du MLLM
$A_{\text{raw}} \leftarrow \textsc{QueryMLLM}(P)$ `// Appel au modèle de langage (LLM/MLLM)`
$A \leftarrow \textsc{PostProcess}(A_{\text{raw}})$ `// Nettoyage / normalisation de la réponse`

Étape 6 : vérification grapho-guidée et explication
$(flag, G_{\text{exp}}) \leftarrow \textsc{VerifyAndExplain}(A, G^*_{\text{RDF}}, \Theta)$ `// Contrôle de cohérence et extraction d'un sous-graphe témoin`

Étape 7 : formatage de la sortie
$R \leftarrow \textsc{FormatAnswer}(A, flag, G_{\text{exp}})$ `// Fusion texte + statut + explication graphe`
return (R, G_{exp})

Algorithme 2 : Vérification grapho-guidée d'une réponse de MLLM et construction d'un sous-graphe d'explication

Input : A : réponse textuelle produite par le MLLM ;
G^*_{RDF} : graphe de connaissances RDF optimisé ;
Θ : paramètres de vérification (seuils θ_{OK}, θ_{PARTIEL}, etc.).
Output : $flag$: statut global de la vérification (`OK`, `PARTIEL`, `KO`) ;
G_{exp} : sous-graphe d'explication extrait de G^*_{RDF}.

Étape 1 : extraction d'entités et de relations à partir du texte
$E \leftarrow$ ExtractEntitiesAndRelations(A) `// NER + extraction relationnelle ou LLM guidé`
if $|E| = 0$ **then**
  $flag \leftarrow$ `KO`
  $G_{\text{exp}} \leftarrow \emptyset$
  return $(flag, G_{\text{exp}})$
end

Étape 2 : vérification de chaque assertion dans le graphe
$G_{\text{exp}} \leftarrow \emptyset$
$nb_{\text{supp}} \leftarrow 0$
$nb_{\text{total}} \leftarrow |E|$
foreach $(s, r, o) \in E$ **do**
  if ExistsTripleOrPath($G^*_{\text{RDF}}, s, r, o$) **then**
    $nb_{\text{supp}} \leftarrow nb_{\text{supp}} + 1$
    $G_{\text{exp}} \leftarrow G_{\text{exp}} \cup$ SubgraphWitness($G^*_{\text{RDF}}, s, r, o$) `// Chemin ou sous-graphe témoin`
  else
    marquer (s, r, o) comme « non supporté »
  end
end

Étape 3 : calcul du score de cohérence
$score \leftarrow nb_{\text{supp}} / nb_{\text{total}}$

Étape 4 : décision sur le statut global
if $score \geq \theta_{\text{OK}}$ **then**
  $flag \leftarrow$ `OK`
else
  if $score \geq \theta_{\text{PARTIEL}}$ **then**
    $flag \leftarrow$ `PARTIEL`
  else
    $flag \leftarrow$ `KO`
  end
end
return $(flag, G_{\text{exp}})$

4.2 Vérification grapho-guidée et construction d'un sous-graphe d'explication

Le module VerifyAndExplain occupe une place centrale dans le dispositif : il permet d'exploiter G^*_{RDF} pour contrôler les réponses du MLLM et fournir une explication structurée Algorithme 2. L'idée est d'extraire des entités et relations de la réponse, de vérifier pour chacune si elle est supportée par le graphe, puis de construire un sous-graphe d'explication à partir des chemins correspondants.

5 Expérimentations et état d'avancement

À ce stade, les travaux réalisés portent principalement sur la *couche symbolique* et sur la mise en place du pipeline neuro-symbolique, tandis que l'intégration complète avec un MLLM et son évaluation systématique sont en cours.

5.1 Scénario T2DM et prototype GraphSynth

Les travaux de thèse Beldi (2024) s'appuient sur un scénario de surveillance d'un patient atteint de diabète de type 2, combinant données de DME (historique médical, examens, prescriptions, notes) et mesures issues de dispositifs médicaux. Un prototype complet *GraphSynth* (frontend React, backend Python) implémente la construction de *DG*, les opérateurs de synthèse et la visualisation des graphes, ainsi qu'une première intégration avec un modèle de langage pour les opérateurs `extract` et `abstract`. Les résultats qualitatifs rapportés dans Beldi et al. (2023) montrent une bonne acceptabilité de ces vues graphiques par des médecins généralistes pour la préparation de consultation T2DM.

5.2 Vers une évaluation neuro-symbolique avec MLLM

L'étape suivante, en cours de mise en œuvre, consiste à évaluer l'apport de la couche symbolique sur un MLLM, en comparant deux conditions : (i) un *baseline MLLM* interrogé avec un contexte purement textuel (documents concaténés) ; (ii) un *MLLM neuro-symbolique* conditionné par un contexte construit à partir de G^*_{RDF} et doté du module de vérification grapho-guidée. Les métriques envisagées incluent : (i) la factualité des réponses évaluée par des cliniciens, (ii) le taux d'hallucinations (assertions non supportées par le graphe), (iii) la couverture explicative (part de la réponse justifiée par un sous-graphe), et (iv) la préférence utilisateur dans un protocole de type A/B. La mise en place de ce protocole et la collecte de résultats constituent un axe de travail en cours.

6 Discussion et perspectives

L'architecture proposée place un *graphe de connaissances centré sur le DME* au cœur du système, construit dynamiquement à partir des données du patient via GraphSynth, DGsumm, Topic Graph Summary et RDF-GraphSyn/HERSE. Ce graphe joue

un double rôle : (i) mémoire symbolique pour construire un contexte RAG structuré destiné au MLLM, (ii) référence factuelle pour vérifier a posteriori les réponses et en extraire un sous-graphe explicatif. Par rapport à des approches fondées uniquement sur des KGs externes ou sur un RAG textuel, le cadre met en avant un *couplage étroit* entre un graphe patient-centrique et le modèle neuronal, ainsi qu'une *personnalisation explicite* via UserProfile-Graph.

Plusieurs limites subsistent néanmoins. L'intégration complète avec un MLLM biomédical et l'évaluation à grande échelle n'ont pas encore été menées ; la qualité de la vérification dépendra fortement des modules d'extraction d'entités et de relations, eux-mêmes imparfaits ; enfin, la complexité d'un déploiement réel (scalabilité, sécurité, confidentialité) reste à étudier. De plus, le cadre présenté est pour l'instant centré sur des données textuelles et structurées, la prise en compte systématique d'images et de signaux relevant encore du travail futur. Les perspectives immédiates portent sur : (i) la mise en œuvre complète du pipeline neuro-symbolique et son évaluation sur plusieurs dossiers T2DM ; (ii) l'extension à d'autres pathologies chroniques ; (iii) l'enrichissement de G^*_{RDF} par des ressources biomédicales externes pour renforcer le contrôle factuel ; (iv) l'intégration progressive d'une multimodalité grapho-guidée (images, signaux). À plus long terme, ce travail s'inscrit dans une vision d'*IA clinique neuro-symbolique* où les MLLMs ne sont pas utilisés seuls, mais enchâssés dans des architectures pilotées par des graphes de connaissances, afin d'améliorer robustesse, traçabilité et explicabilité pour les professionnels de santé.

Références

Agrawal, G., T. Kumarage, Z. Alghamdi, et H. Liu (2024). Can knowledge graphs reduce hallucinations in llms ? : A survey. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies (Volume 1 : Long Papers)*, pp. 3947–3960.

Alsentzer, E., J. Murphy, W. Boag, W.-H. Weng, D. Jindi, T. Naumann, et M. McDermott (2019). Publicly available clinical bert embeddings. In *Proceedings of the 2nd clinical natural language processing workshop*, pp. 72–78.

Beldi, A. (2024). *Synthèse Graphique Multidimensionnelle : Application aux documents hétérogènes*. Ph. D. thesis, Université de Pau et des Pays de l'Adour ; Université de Tunis El Manar.

Beldi, A., J. R. Richa, S. Sassi, R. Chbeir, et A. Jemai (2023). A novel approach for extracting summarized rdf graph from heterogeneous corpus. In *2023 International Conference on Innovations in Intelligent Systems and Applications (INISTA)*, pp. 1–7. IEEE.

Beldi, A., S. Sassi, R. Chbeir, et A. Jemai (2022b). The current state of summarization and visualization in electronic health record (ehr) based on ehr interoperability. *Medical Information Processing and Security : Techniques and applications*, 87–123.

Beldi, A., S. Sassi, R. Chbeir, et A. Jemai (2022c). Schema formalism for semantic summary based on labeled graph from heterogeneous data. In *Asian Conference on Intelligent Information and Database Systems*, pp. 27–44. Springer.

Beldi, A., S. Sassi, R. Chbeir, et A. Jemai (2023). Dg_summ : A schema-driven approach for personalized summarizing heterogeneous data graphs. *Computer Science and Information Systems 20*(4), 1591–1638.

Beldi, A., S. Sassi, R. Chbeir, et A. Jemai (2024). Herse : Handling and enhancing rdf summarization through blank node elimination. In *International Symposium on Methodologies for Intelligent Systems*, pp. 87–101. Springer.

Beldi, A., S. Sassi, et A. Jemai (2022a). Learn2sum : A new approach to unsupervised text summarization based on topic modeling. In *Proceedings of the 14th international conference on management of digital ecosystems*, pp. 136–143.

Culié, D. (2024). *Du dossier médical à l'utilisation des données. Élaboration de procédés basés sur l'intelligence artificielle pour extraire, structurer et analyser les données en pathologie thyroïdienne*. Ph. D. thesis, Université Côte d'Azur.

Devlin, J., M.-W. Chang, K. Lee, et K. Toutanova (2019). Bert : Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics : human language technologies, volume 1 (long and short papers)*, pp. 4171–4186.

Gu, Y., S. Zhang, N. Usuyama, Y. Woldesenbet, C. Wong, P. Sanapathi, M. Wei, N. Valluri, E. Strandberg, T. Naumann, et al. (2023). Distilling large language models for biomedical knowledge extraction : A case study on adverse drug events. *arXiv preprint arXiv :2307.06439*.

Hogan, A., E. Blomqvist, M. Cochez, C. d'Amato, G. D. Melo, C. Gutierrez, S. Kirrane, J. E. L. Gayo, R. Navigli, S. Neumaier, et al. (2021). Knowledge graphs. *ACM Computing Surveys (Csur) 54*(4), 1–37.

Huang, L., W. Yu, W. Ma, W. Zhong, Z. Feng, H. Wang, Q. Chen, W. Peng, X. Feng, B. Qin, et al. (2025). A survey on hallucination in large language models : Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems 43*(2), 1–55.

Kondylakis, H., D. Kotzinos, et I. Manolescu (2019). Rdf graph summarization : principles, techniques and applications (tutorial). In *EDBT/ICDT 2019-22nd International Conference on Extending Database Technology-Joint Conference*.

Lewis, P., E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, et al. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems 33*, 9459–9474.

Liu, Y., T. Safavi, A. Dighe, et D. Koutra (2018). Graph summarization methods and applications : A survey. *ACM computing surveys (CSUR) 51*(3), 1–34.

Schulz, S., E. Beisswanger, L. van den Hoek, O. Bodenreider, et E. M. van Mulligen (2009). Alignment of the umls semantic network with biotop : methodology and assessment. *Bioinformatics 25*(12), i69–i76.

Singhal, K., S. Azizi, T. Tu, S. S. Mahdavi, J. Wei, H. W. Chung, N. Scales, A. Tanwani, H. Cole-Lewis, S. Pfohl, et al. (2023). Large language models encode clinical

knowledge. *Nature 620*(7972), 172–180.

Summary

Large multimodal language models (MLLMs) offer promising capabilities for summarizing and analyzing Electronic Health Records (EHRs). However, their deployment in clinical settings remains limited by hallucinations, lack of reasoning traceability, and insufficient integration of structured knowledge from graphs or ontologies.

In this paper, we propose a neuro-symbolic architecture for EHR summarization based on two main components developed in a PhD thesis: (i) *GraphSynth*, which models heterogeneous EHR data as a data graph and summarizes it using a family of dedicated operators, and (ii) *RDF-GraphSyn*, which translates these summary graphs into optimized RDF knowledge graphs.

These graphs constitute a symbolic layer used to condition an MLLM within a *Retrieval-Augmented Generation* (RAG) framework and to control its outputs through graph-guided verification. We show how this framework can produce more factual and explainable answers, each associated with a supporting evidence subgraph. A use case focusing on the summarization of a type 2 diabetes patient record illustrates the proposed approach.

Keywords: Electronic Health Records, knowledge graphs, neuro-symbolic AI, large language models, RAG, explainability.