Graphe de connaissances : représentation, raisonnement et apprentissage

Adrien Coulet, équipe HeKA

Inria

Inserm
La science pour la santé
From science to health

Université Paris Cité



2 façons de concevoir le monde

2 façons d’acquérir des connaissances

Plan

1. Définitions
   a) Distinction entre données, information et connaissances
   b) Définitions d'un graphe de connaissances
   c) Les types de graphes de connaissances
2. Le raisonnement sur les graphes de connaissances
   a) Raisonnement RDF
   b) Raisonnement OWL
3. **L'apprentissage sur les graphes de connaissances**
   a) Approches classiques
   b) Approches par marches aléatoires
   c) Approches translationnelles
   d) GCN
   e) GAT

Plan

1. Définitions
   a) Distinction entre données, information et connaissances
   b) Définitions d'un graphe de connaissances
   c) Les types de graphes de connaissances
2. Le raisonnement sur les graphes de connaissances
   a) Raisonnement RDF
   b) Raisonnement OWL
3. **L'apprentissage sur les graphes de connaissances**
   a) Approches classiques
   b) Approches par marches aléatoires
   c) Approches translationnelles
   d) GCN
   e) GAT





Des données aux connaissances

- **Données :** ensemble de signes et symboles
- **Informations :** données + sens (ou sémantique)
- **Connaissances :** information + possibilité de mise en action de l'information interprétée

- "Dobrý den, pane. Kolik je hodin?"
- "Je přesně 08:32."

- "Bonjour Monsieur. Quelle heure est-il ?"
- "Il est précisément 8h32 ."

- "Bonjour Monsieur. Quelle heure est-il ?"
- "Il est précisément 8h32 ."

- "J'ai un rendez vous prévu à 8h30. Je suis en retard. Il faut que je cours."



- "Bonjour Monsieur. Quelle heure est-il ?"
- "Il est précisément 8h32 ."

```
1: situation= "à l'heure",
2: vitesse_transport = "marche",
3: heure_rdv = 8h30,
4: heure_actuelle = 8h32,
5: si heure_actuelle > heure_rdv alors situation = "en retard",
6: si situation = »en retard"
7: alors vitesse_transport = "cours"
8: et parole="J'ai un rendez vous prévu à "+heure_rdv+". Je suis "+situation+". Il faut que je "+vitesse_transport+".".
```

- "J'ai un rendez vous prévu à 8h30. Je suis en retard. Il faut que je cours."



Graphe de connaissances : définitions

- Définition générale :

  « Un graphe avec des connaissances » (Paulheim, 2017)

- Définition consensuelle :

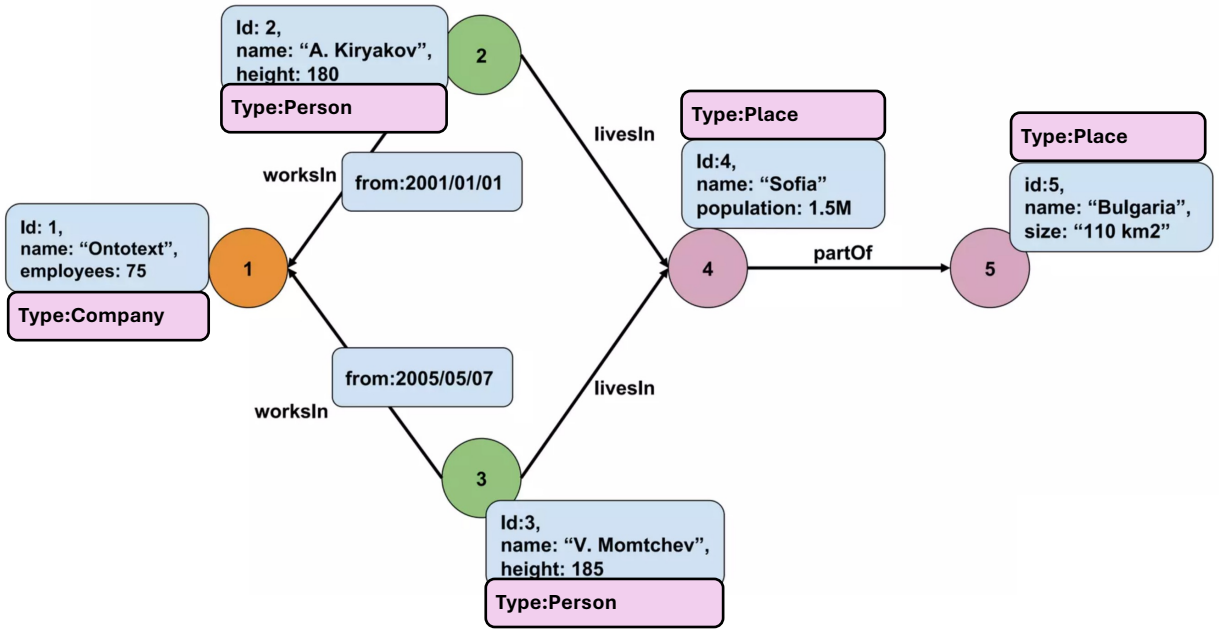
  « Un graphe qui cherche à représenter des connaissances et dont les nœuds représentent des entités du monde et les arcs des relations entre ces entités » (Hogan *et al.*, 2021)

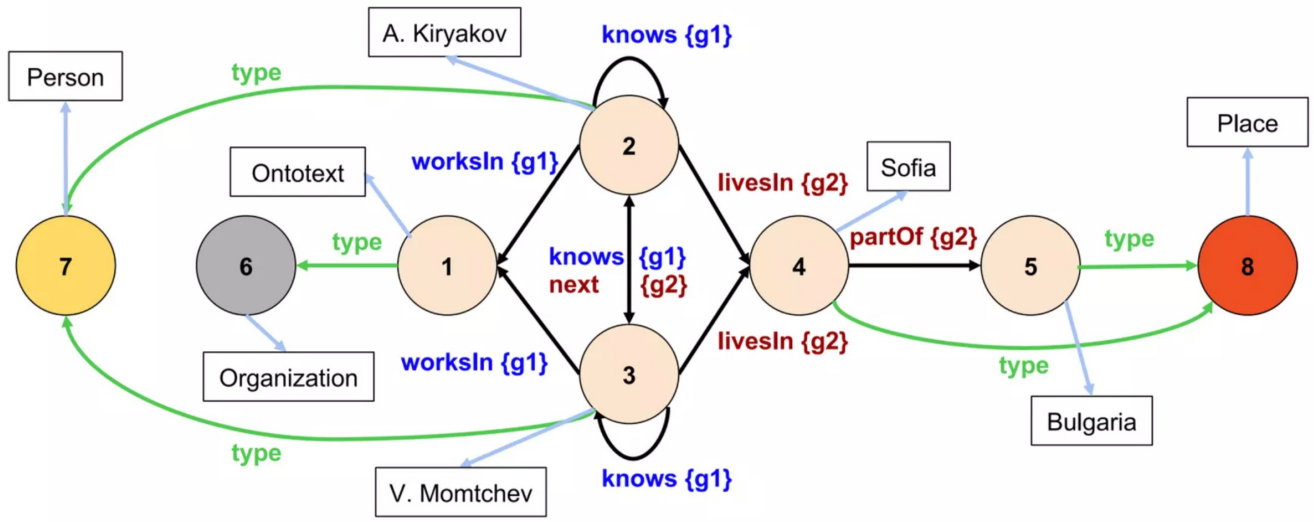
- Définition Web sémantique :

  « Un graphe qui intègre des connaissances diverses dans l'objectif de pouvoir en dériver de nouvelles » (Ehrlinger and Wöß, 2016)

- Différents types de graphes de connaissances
  - Les graphes à propriétés ou graphes attribués (*Property graphs)*
  - Les graphes RDF
  - D'autres : les hypergraphes, …

Graphe à propriétés *vs.* Graphe RDF

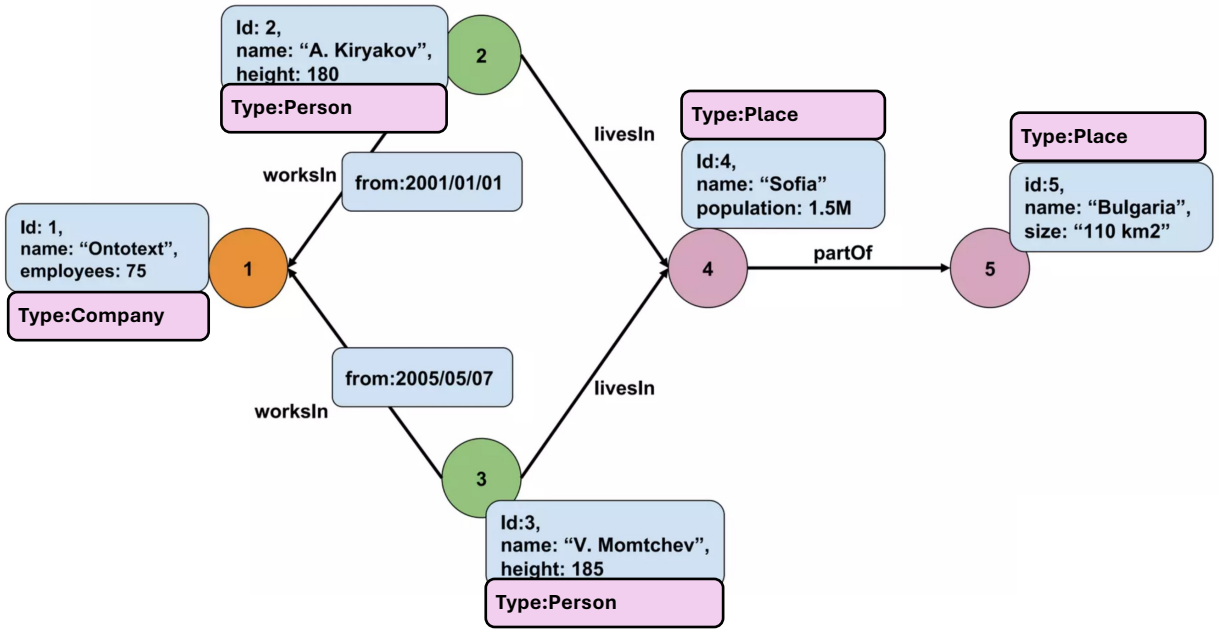


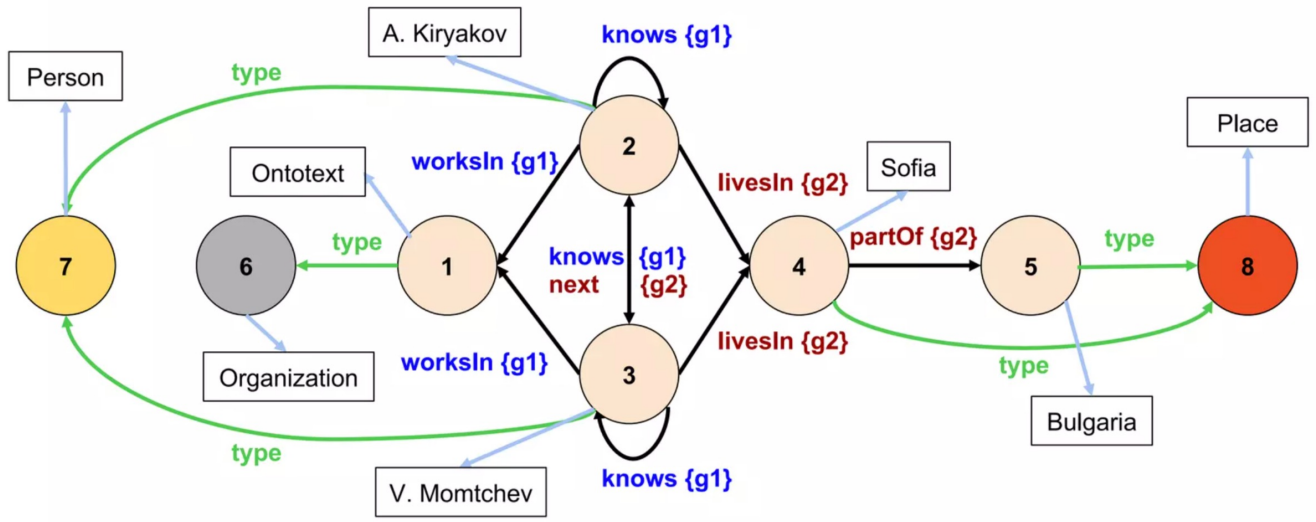


Graphe à propriétés :
Multi-graphe dirigé, étiqueté, cyclique, typé

Graphe RDF :
Multi-graphe dirigé, étiqueté, cyclique, typé

Graphe à propriétés *vs.* Graphe RDF





Différences

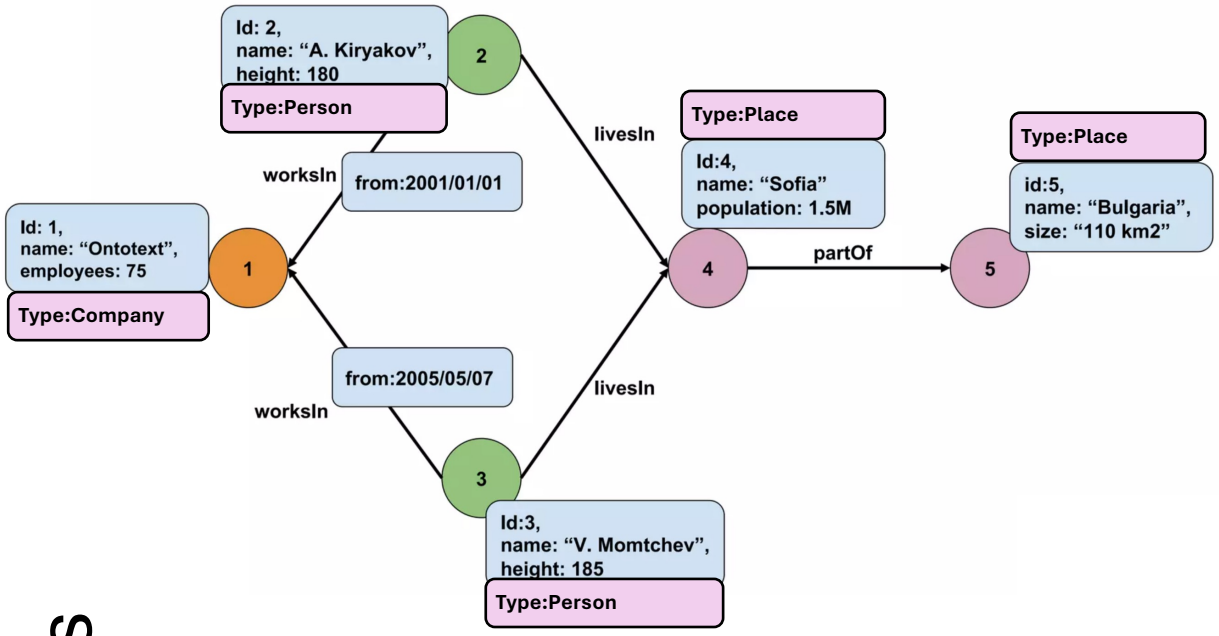
Graphe à propriétés :
Multi-graphe dirigé, étiqueté, cyclique, typé

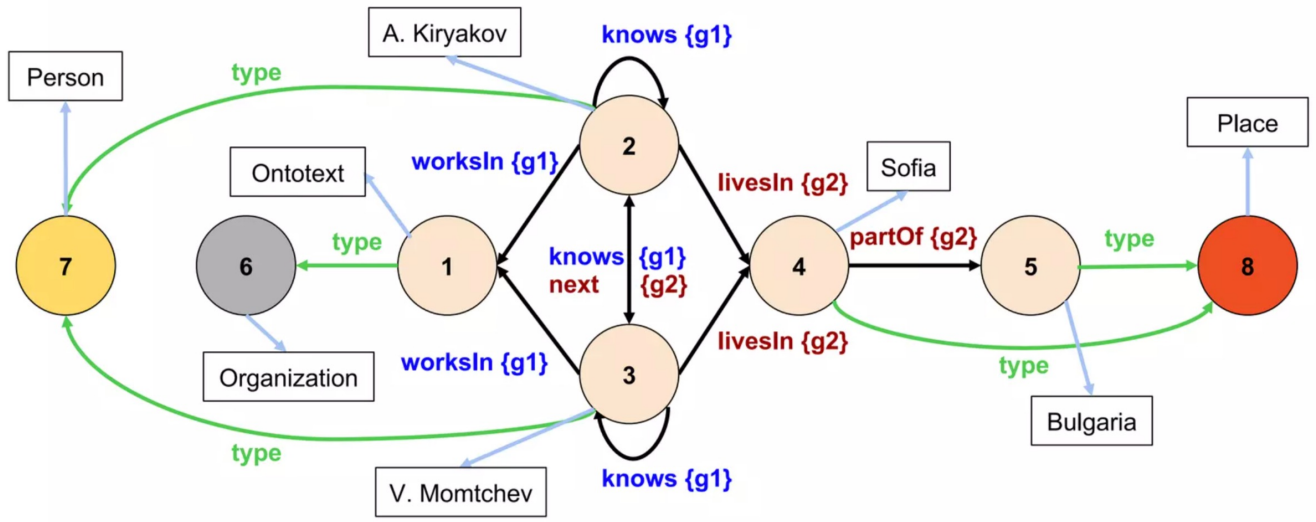
- Nœuds et arcs sont associés à des ensembles de propriétés *i.e.*, des paires (clé,valeur)
- Pas de standard particulier pour les noms de nœuds, arcs et leurs propriétés
- Pas de langage standard d'interrogation (Cypher, GQL, etc.)
- Pas de langage de schéma ou de sémantique formelle associée

Graphe RDF :
Multi-graphe dirigé, étiqueté, cyclique, typé

- Tout est représenté sous forme de nœuds et d'arc
- Standards de nommage par le W3C
- Langage standard d'interrogation: SPARQL
- Langages de schéma (RDFS, OWL) et sémantique formelle associée, autorisant le raisonnement

Graphe à propriétés *vs.* Graphe RDF





Pros & Cons

Graphe à propriétés :

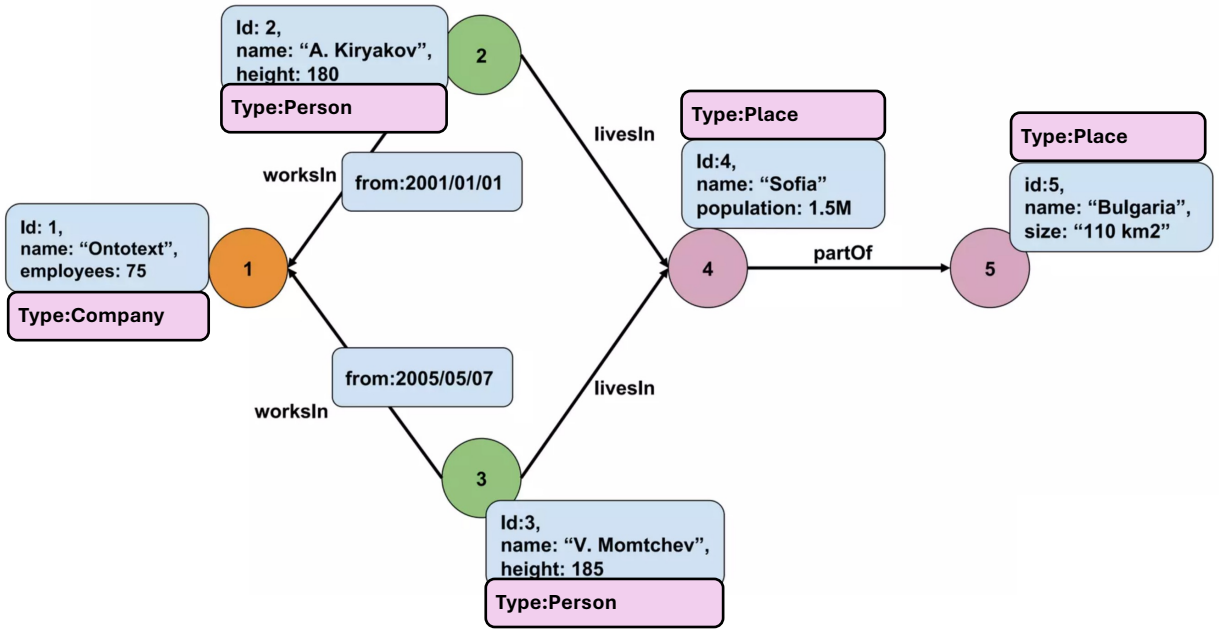
- Modèle plus simple, plus compact
- Adapté à l'analyse de données
- Passe plus facilement à l'échelle de très grands graphes

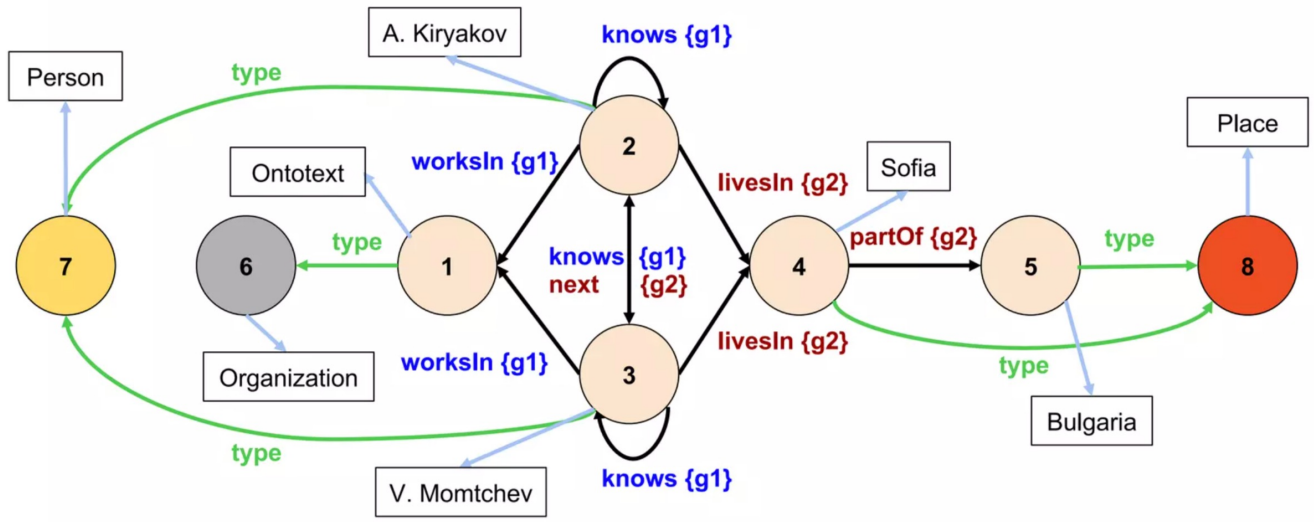
Graphe RDF :

- Modèle plus expressif et associé à des standards
- Adapté à la représentation de connaissances, à l'échange et l'interconnection de données, à l'integration de données
- Plus "FAIR"

Source : https://www.ontotext.com/knowledgehub/fundamentals/rdf-vs-property-graphs/

Graphe à propriétés *vs.* Graphe RDF





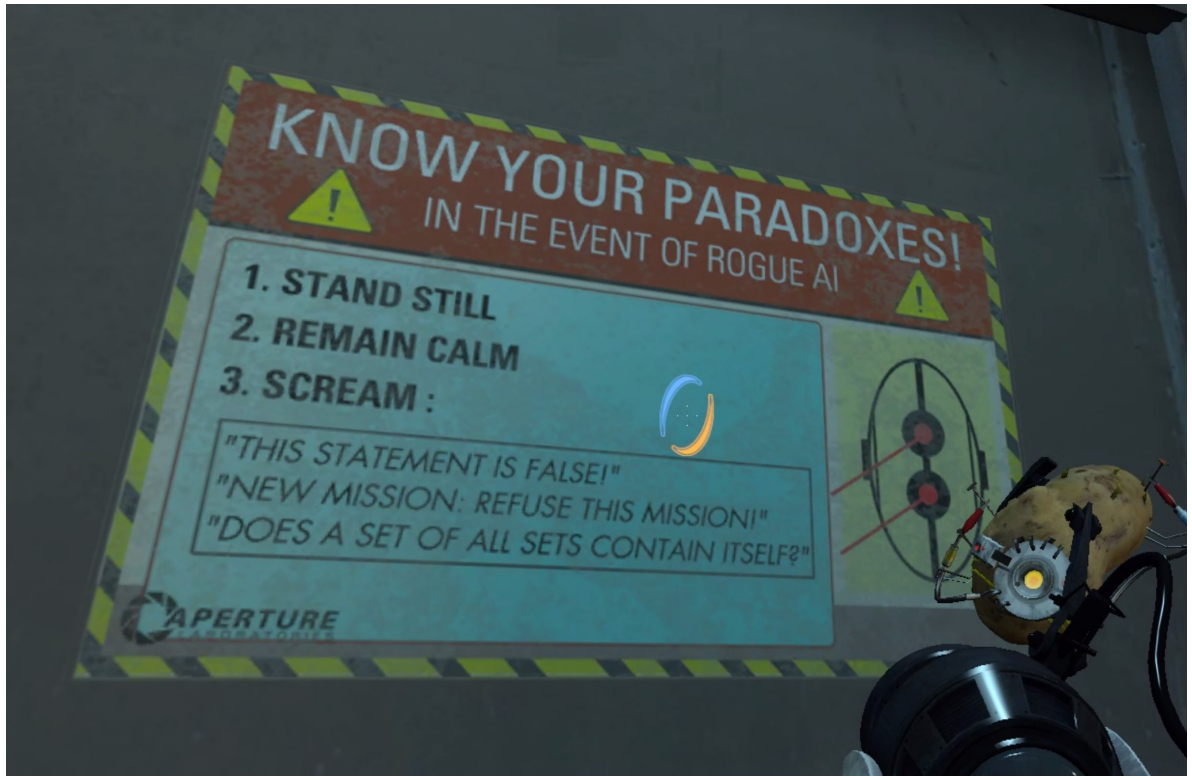
Outils

- Neo4j
- ArangoDB
- TigerGraph
- Oracle Property Graph
- …

- Virtuoso, OpenLink
- AllegroGraph
- GraphDB, OntoText
- Oracle RDF
- …

- Amazone Neptune
- …

Source : https://www.ontotext.com/knowledgehub/fundamentals/rdf-vs-property-graphs/



Raisonnement avec RDF / RDFS

Raisonnement avec RDFS

Les règles de déduction

$$\frac{r \text{ rdfs:domain } x \;.\qquad u\ r\ y\ .}{u \text{ rdf:type } x\ .}$$

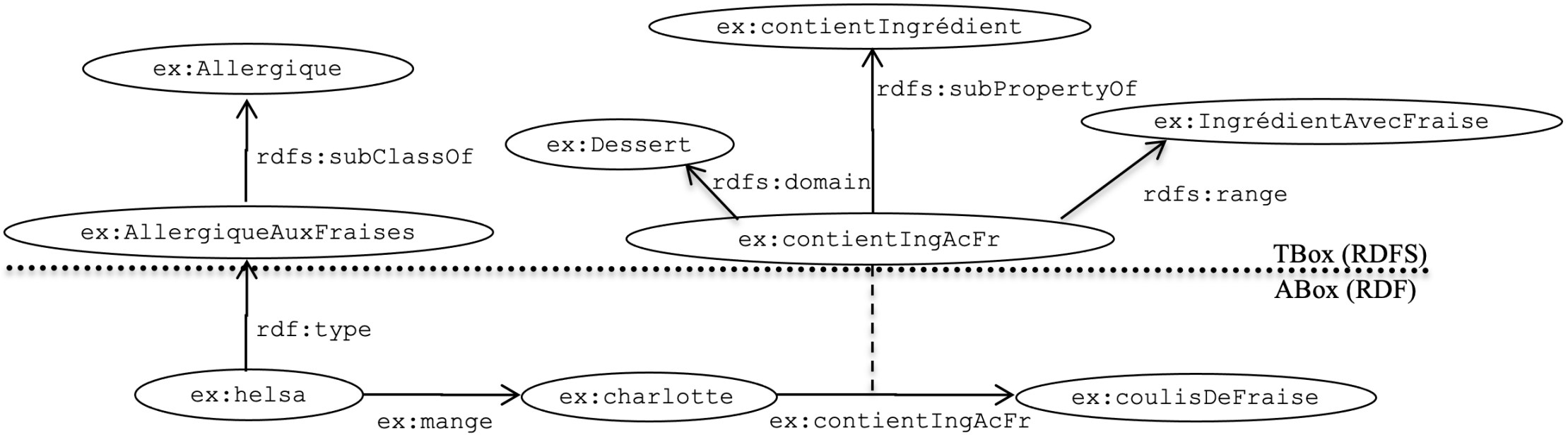
La même règle de déduction existe pour l'objet de *rdfs:range*.

$$\frac{x1 \text{ rdfs:subClassOf } x2\ .\qquad u\ rdf{:}type\ x1\ .}{u\ rdf{:}type\ x2\ .}$$

$$\frac{u \text{ rdfs:subClassOf } v\ .\quad v \text{ rdfs:subClassOf } x\ .}{u \text{ rdfs:subClassOf } x\ .}$$

Le même type de règle est défini pour *rdfs:subPropertyOf*

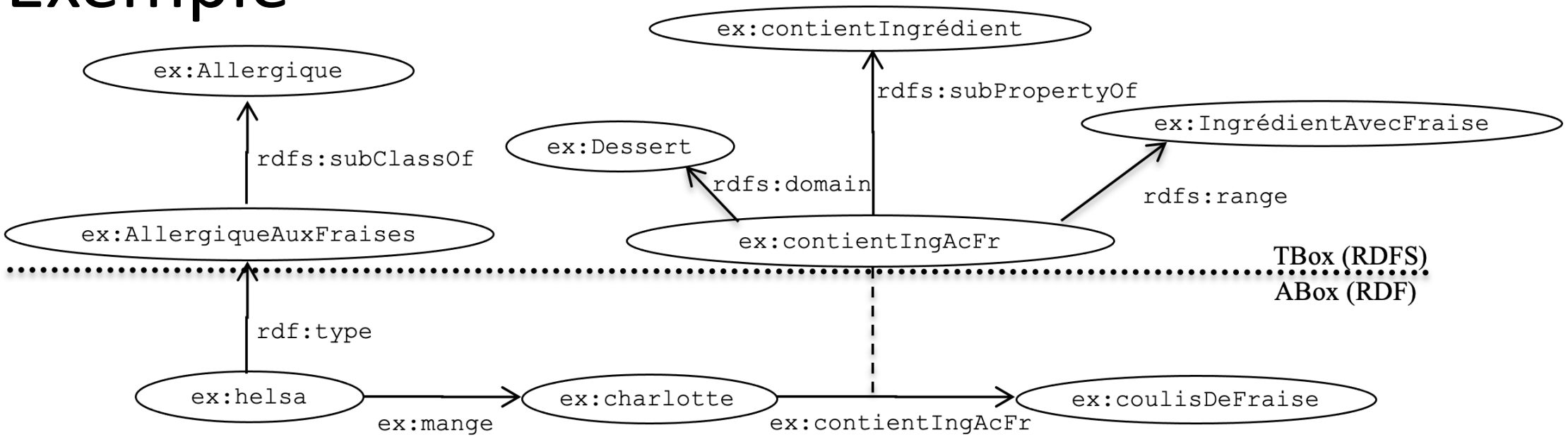
Exemple



G_1

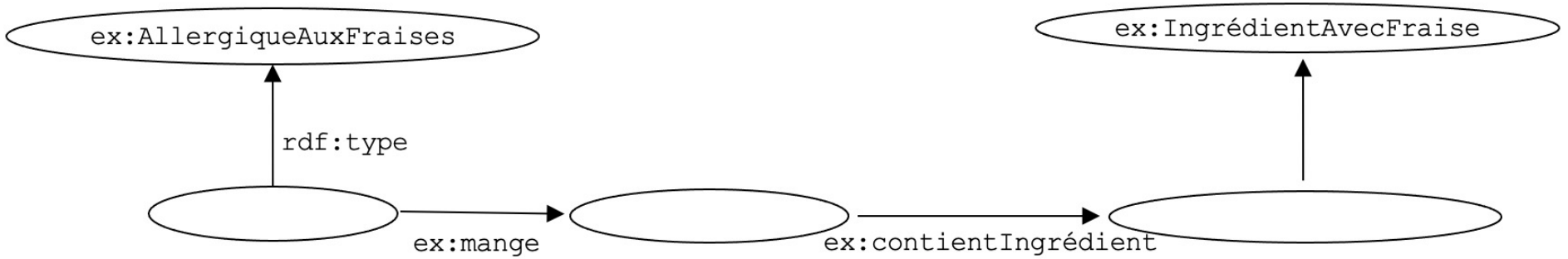
Quelqu'un qui est allergique aux fraises va t-il manger un plat contenant des ingrédients à base de fraise ?

Exemple



G_1

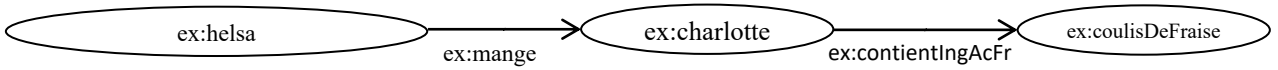
Est ce que quelqu'un qui est allergique aux fraises va manger un plat contenant des ingrédients à base de fraise ?



G_2

$$G_1 \vDash G_2 ?$$

Exemple



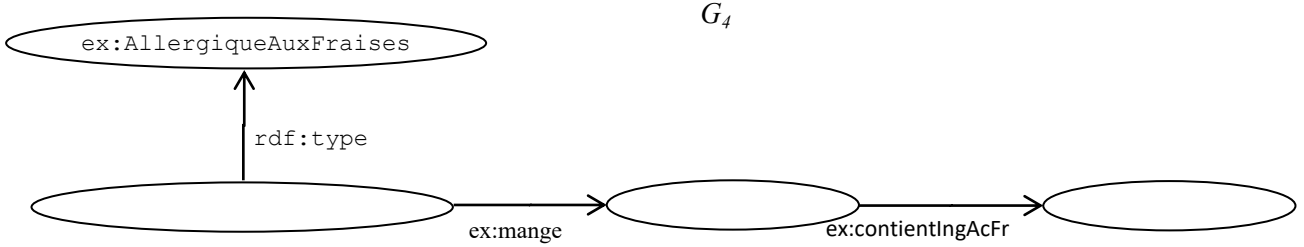
G_1



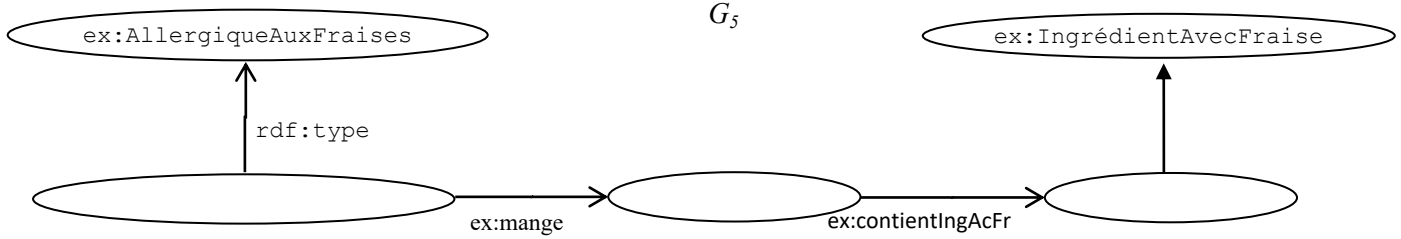
G_3



G_4



G_5

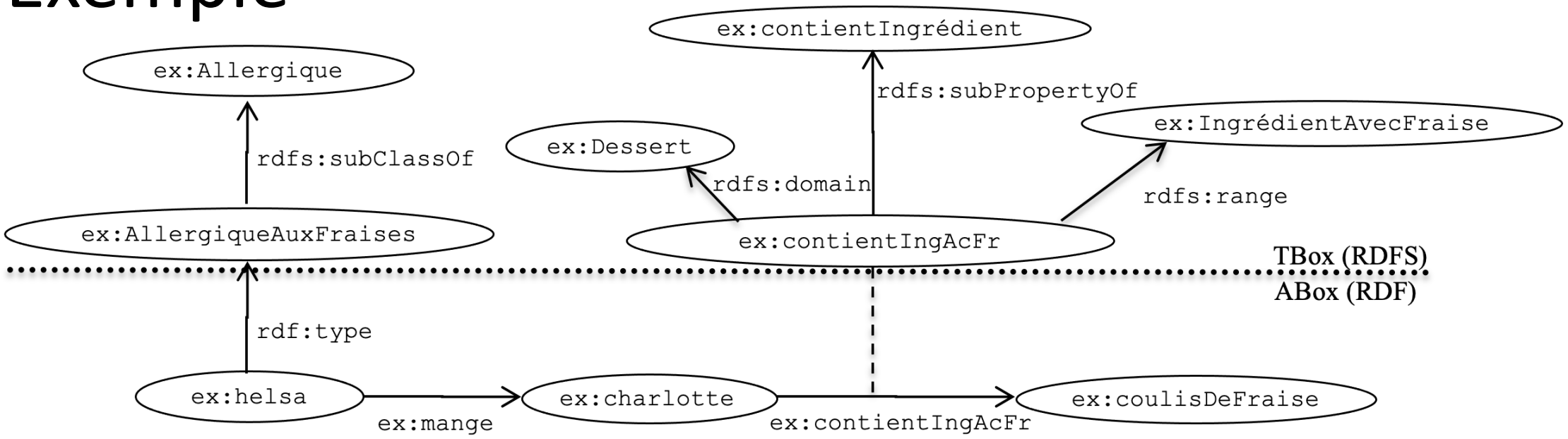


G_6

• • •

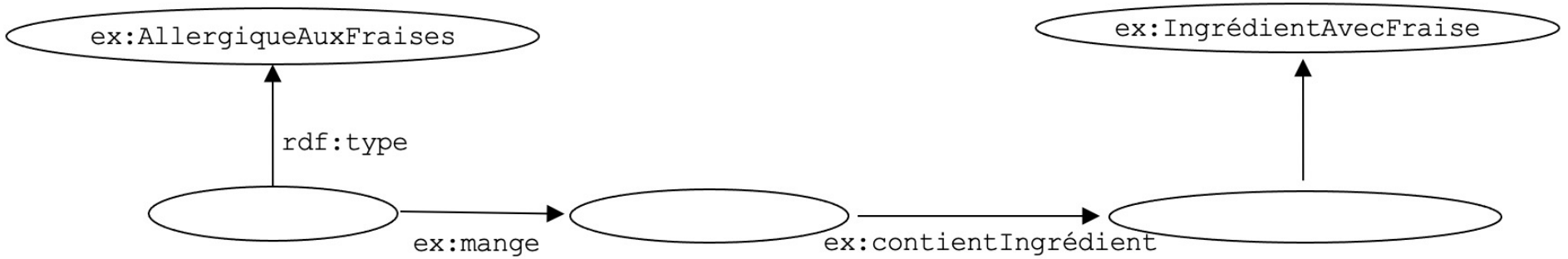
$G_1 \models G_3,\ G_3 \models G_4,\quad G_4 \models G_5,\ \ldots$

Exemple



G_1

Est ce que quelqu'un qui est allergique aux fraises va manger un plat contenant des ingrédients à base de fraise ?



G_2

$G_1 \models G_2$

OWL



Les différents mécanismes de raisonnement disponibles

- Tests de satisfaisabilité

$$BC \not\models C \equiv \bot$$

- Test de subsomption

$$BC \models C \sqsubseteq D$$

- Test d'instanciation

$$BC \models C(a)$$

Les différents mécanismes de raisonnement disponibles

- Vérification de cohérence de la Base de Connaissances
- Classification des concepts
- Recherche d’instances
- Classification d’instance
- Recherche du subsumant le plus spécifique

Plan

1. Définitions
   a) Distinction entre données, information et connaissances
   b) Définitions d'un graphe de connaissances
   c) Les types de graphes de connaissances
2. Le raisonnement sur les graphes de connaissances
   a) Raisonnement RDF
   b) Raisonnement OWL
3. **L'apprentissage sur les graphes de connaissances**
   a) Approches classiques
   b) Approches par marches aléatoires
   c) Approches translationnelles
   d) GCN
   e) GAT

Les différentes tâches de prédiction sur les graphes

Tâche	Sortie attendue	Exemple
Link prediction	Score ou classement de liens ou triplets	(drug_A, has_targets, protein_P) : true or false (drug_A, has_targets, ?) : classer les cibles candidates d'un médicament
Edge classification / relation prediction	Type de relation pour une paire donnée	(drug_A, ?, disease_B) : treats? causes? (drug_A, treats, disease_B) : true or false
Node classification	Une (ou plusieurs) étiquettes par nœud	Fonctions d'une protéine P (protein_P, has_function,?) (protein_P, rdf:type,?)
Node regression	Une valeur continue pour un noeud	Estimer la demi-vie d'une molécule
Subgraph classification	Une étiquette ou une valeur pour un sous-graphe	Graphe moléculaire : toxicité ou solubilité

> Important à bien specifier pour éviter les ambiguités

Les différentes tâches de prédiction sur les graphes

Tâche	Sortie attendue	Exemple
Link prediction	Score ou classement de liens ou triplets	(drug_A, has_targets, protein_P) : true or false (drug_A, has_targets, ?) : classer les cibles candidates d'un médicament
Edge classification / relation prediction	Type de relation pour une paire donnée	(drug_A, ?, disease_B) : treats? causes? (drug_A, treats, disease_B) : true or false
Node classification	Une (ou plusieurs) étiquettes par nœud	Fonctions d'une protéine P (protein_P, has_function,?) (protein_P, rdf:type,?)
Node regression	Une valeur continue pour un noeud	Estimer la demi-vie d'une molécule
Subgraph classification	Une étiquette ou une valeur pour un sous-graphe	Graphe moléculaire : toxicité ou solubilité

> Important à bien specifier pour éviter les ambiguités

Btw, est-ce que les noeuds ou types de relations ont déjà été vues dans l'ensemble d'entrainement ?
Sont-elles dans le graphe initial ?

Les différentes tâches de prédiction sur les graphes

Tâche	Sortie attendue	Exemple
Link prediction	Score ou classement de liens ou triplets	(drug_A, has_targets, protein_P) : true or false (drug_A, has_targets, ?) : classer les cibles candidates d'un médicament
Edge classification / relation prediction	Type de relation pour une paire donnée	(drug_A, ?, disease_B) : treats? causes? (drug_A, treats, disease_B) : true or false
Node classification	Une (ou plusieurs) étiquettes par nœud	Fonctions d'une protéine P (protein_P, has_function,?) (protein_P, rdf:type,?)
Node regression	Une valeur continue pour un noeud	Estimer la demi-vie d'une molécule
Subgraph classification	Une étiquette ou une valeur pour un sous-graphe	Graphe moléculaire : toxicité ou solubilité

> Important à bien specifier pour éviter les ambiguités

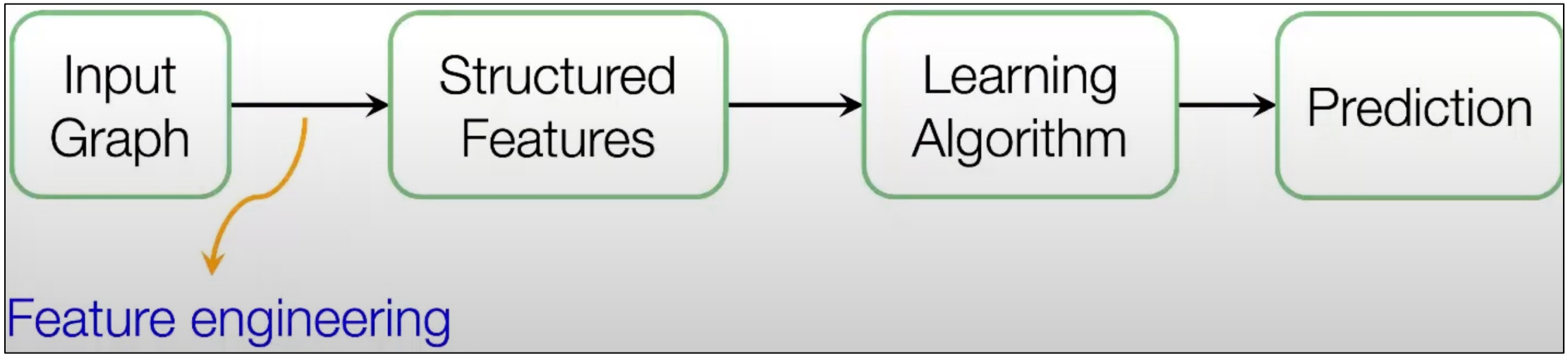
Btw, est-ce que les noeuds ou types de relations ont déjà été vues dans l'ensemble d'entrainement ?
Sont-elles dans le graphe initial ?

> Distinction entre apprentissage **transductif** (le noeud à prédire est présent dans le *train set*)
> et apprentissage **inductif** (le noeud est absent du *train set*)

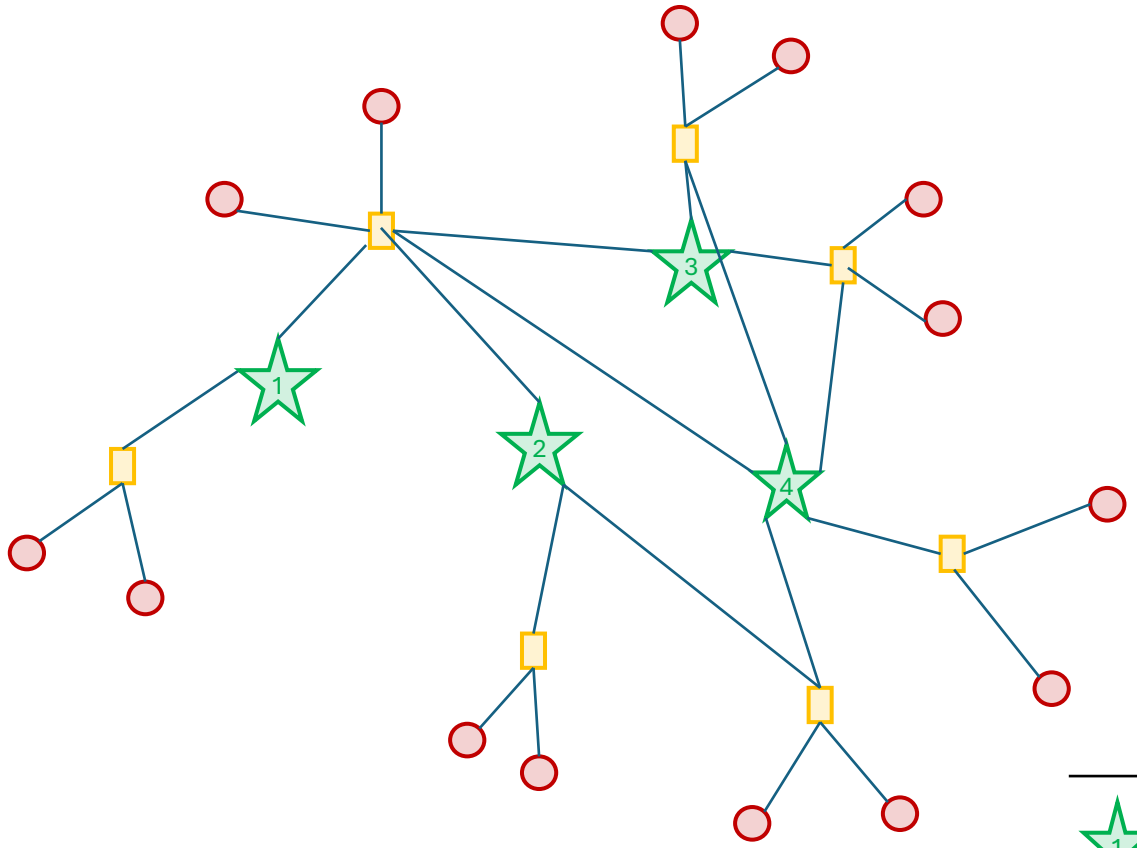
Les différentes tâches de prédiction sur les graphes

Tâche	Sortie attendue	Exemple
Link prediction	Score ou classement de liens ou triplets	(`drug_A, has_targets, protein_P`) : true or false (`drug_A, has_targets, ?`) : classer les cibles candidates d'un médicament
Edge classification / relation prediction	Type de relation pour une paire donnée	(drug_A, ?, disease_B) : treats? causes? (drug_A, treats, disease_B) : true or false
Node classification	Une (ou plusieurs) étiquettes par nœud	Fonctions d'une protéine P (protein_P, has_function,?) (protein_P, rdf:type,?)
Node regression	Une valeur continue pour un noeud	Estimer la demi-vie d'une molécule
Subgraph classification	Une étiquette ou une valeur pour un sous-graphe	Graphe moléculaire : toxicité ou solubilité

> On peut également faire de la resolution d'entité : `drug_A owl:same_as drug_B`
> De la recherche de similarité : top 10 des medicaments les plus similaires à drug_A: `drug_A skos:close_match drug_B`
> Du clustering



Source : Jure Leskovec, CS224W, Stanford



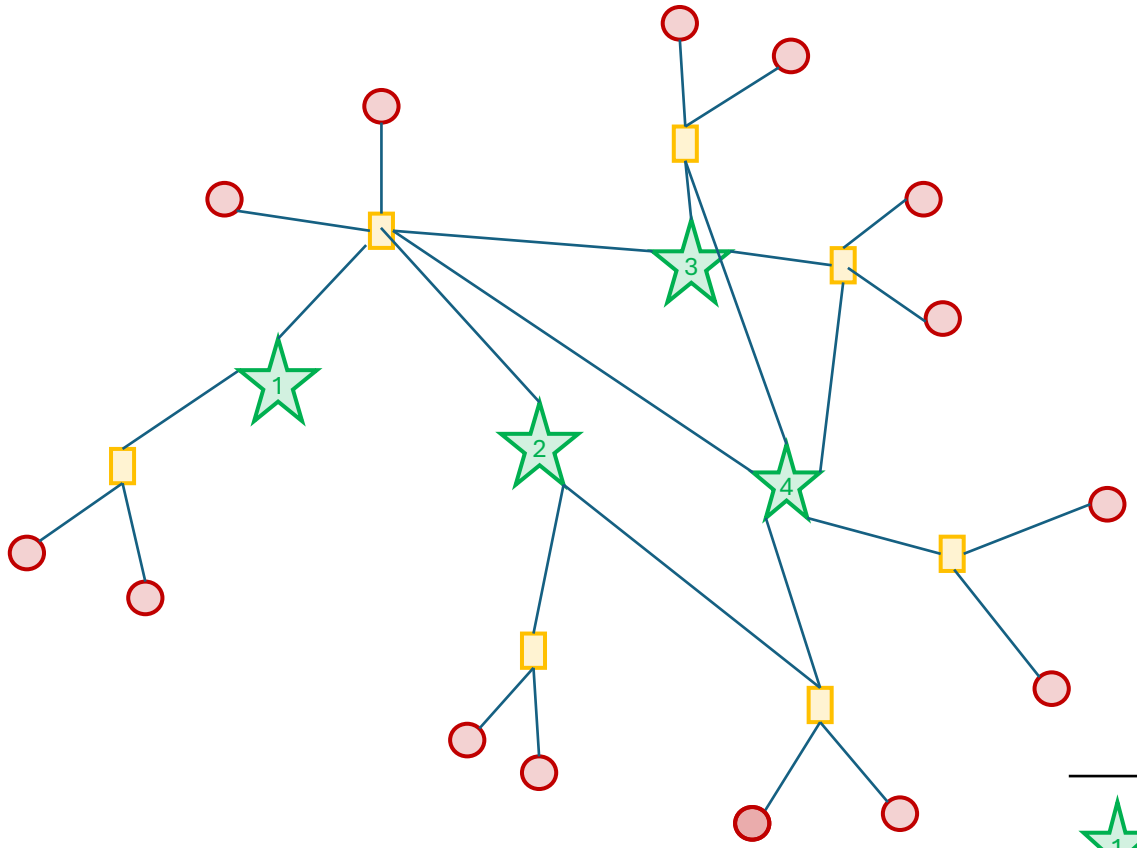
Node features

1	
2	
3	
4	

1 Target node

Neighbourhood = 1

Neighbourhood = 2



Target node

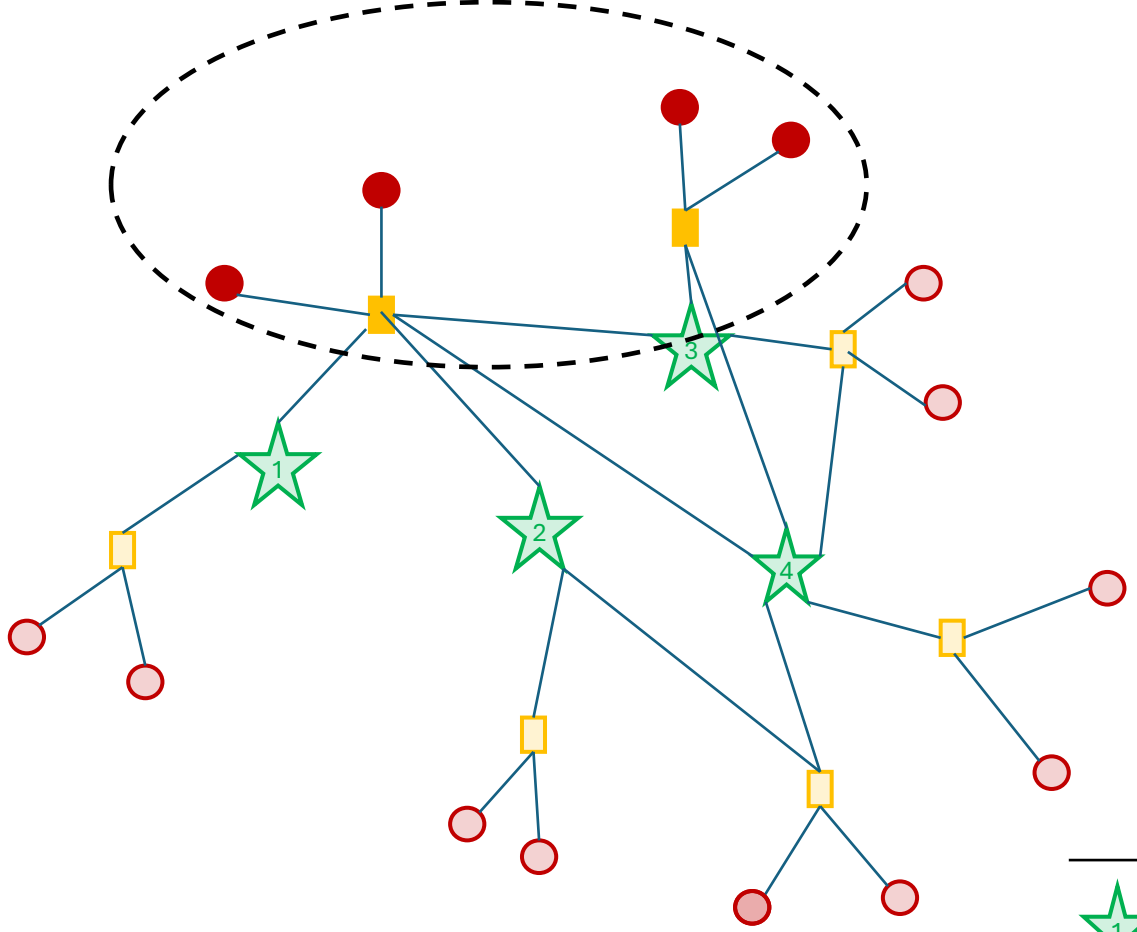
Neighbourhood = 1

Neighbourhood = 2

Node features

Bag of nodes

1	
2	
3	
4	



1 Target node

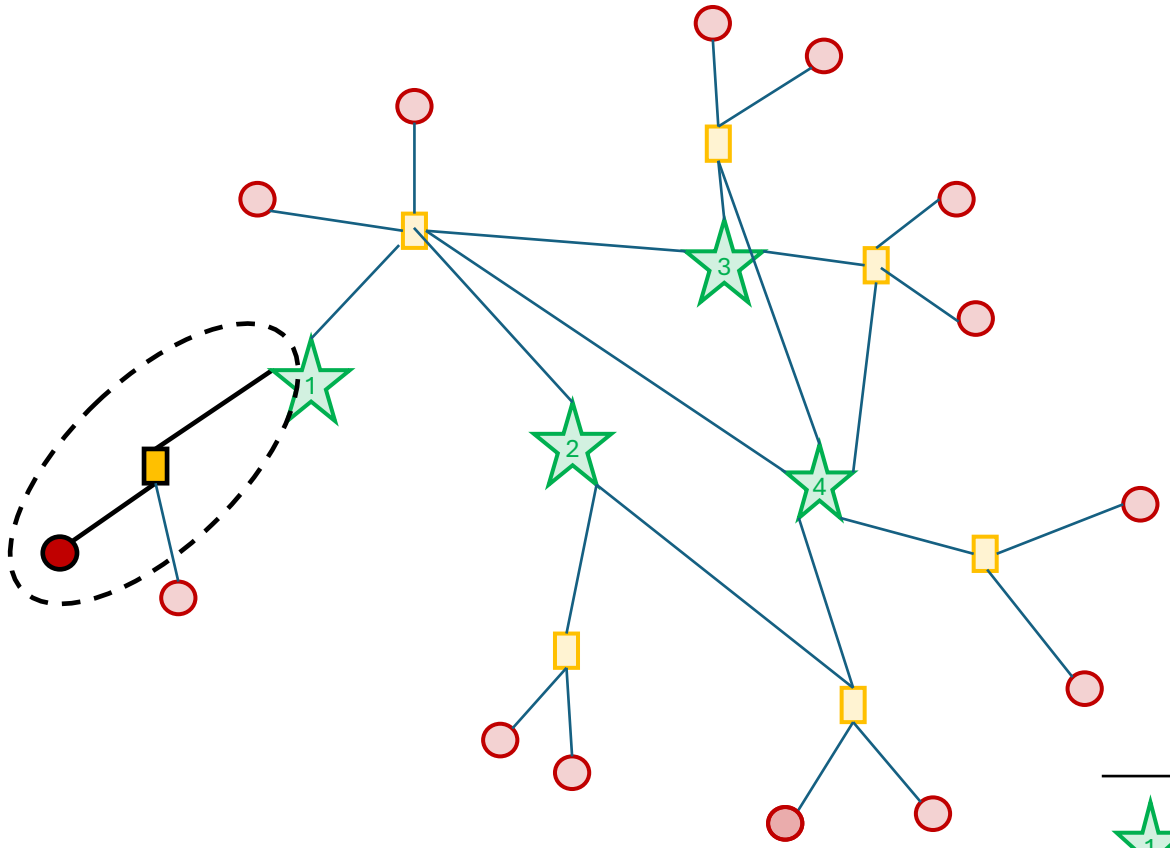
Neighbourhood = 1

Neighbourhood = 2

Node features

Bags of nodes

		...	
1			
2			
3	X		
4	X		



Target node

Neighbourhood = 1

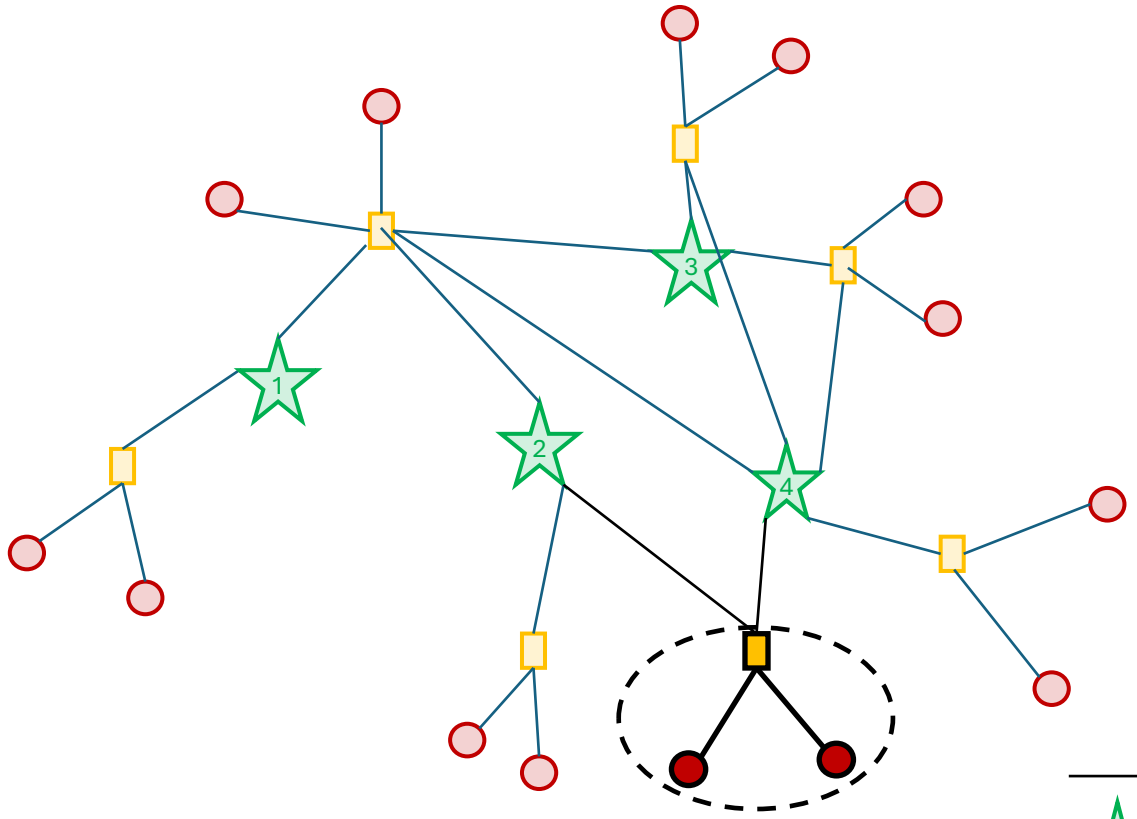
Neighbourhood = 2

Node features

Bags of nodes

Paths

1		X	
2			
3	X		
4	X		



1 Target node

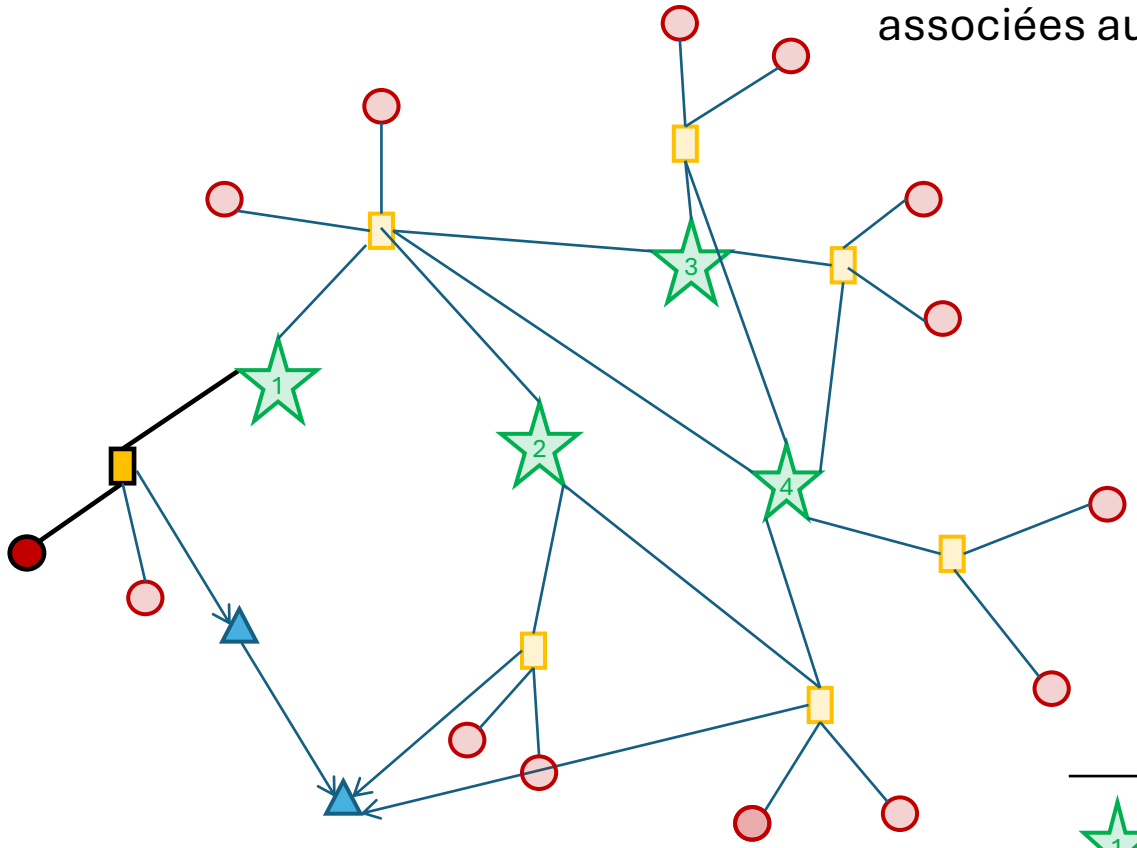
Neighbourhood = 1

Neighbourhood = 2

Node features

	Bags of nodes	Paths	Trees
1		X	
2			X
3	X		
4	X		X

Il est également possible d'utiliser les representations de connaissances associées au KG pour construire des *Bags*, *Path* ou *Tree* généralisés



1 Target node

Neighbourhood = 1

Neighbourhood = 2

Parent of Neighbourhood = 1

Node features

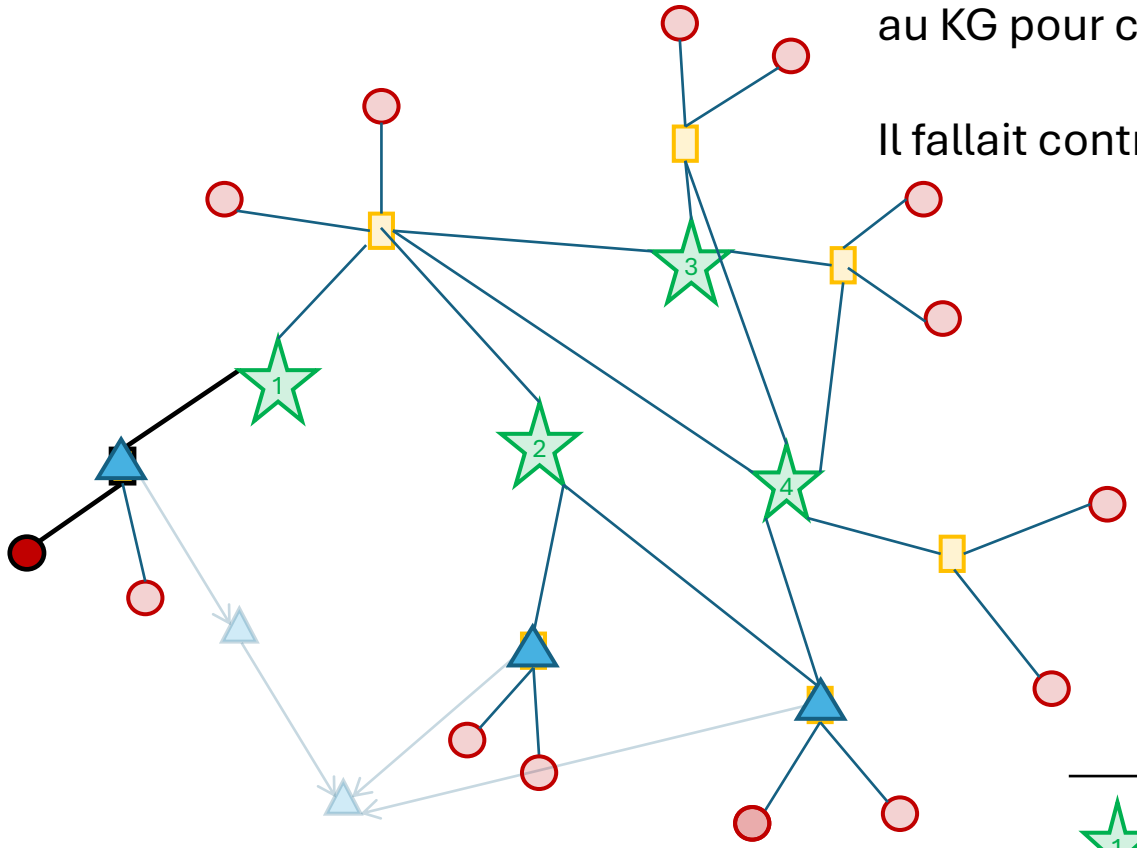
Bags of nodes

Generalized paths

	...	...	
1		X	
2			
3	X		
4	X		

Il est également possible d'utiliser les representations de connaissances associ au KG pour construire des *Bags*, *Path* ou *Tree* généralisés

Il fallait controller l'explosion combinatoire



1 Target node

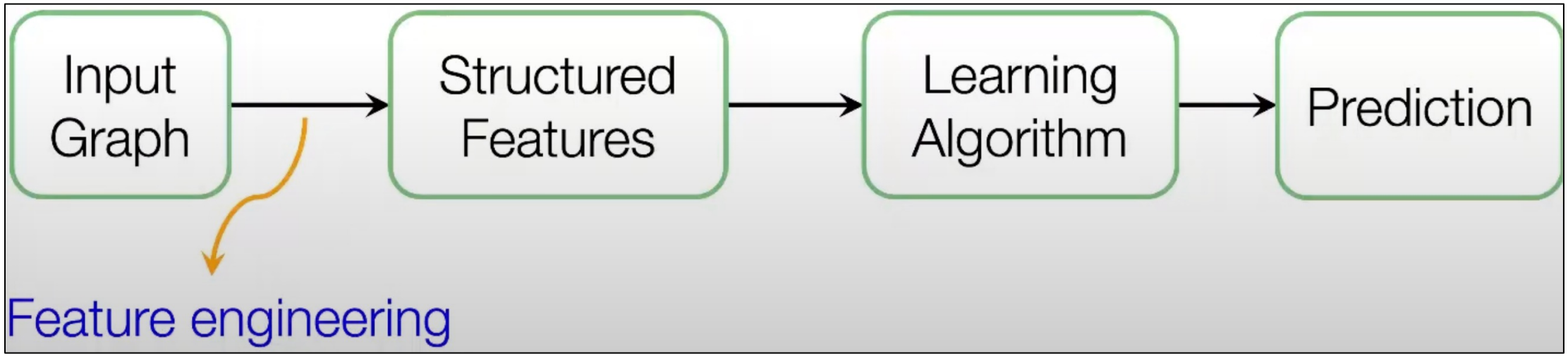
Neighbourhood = 1

Neighbourhood = 2

Parent of Neighbourhood = 1

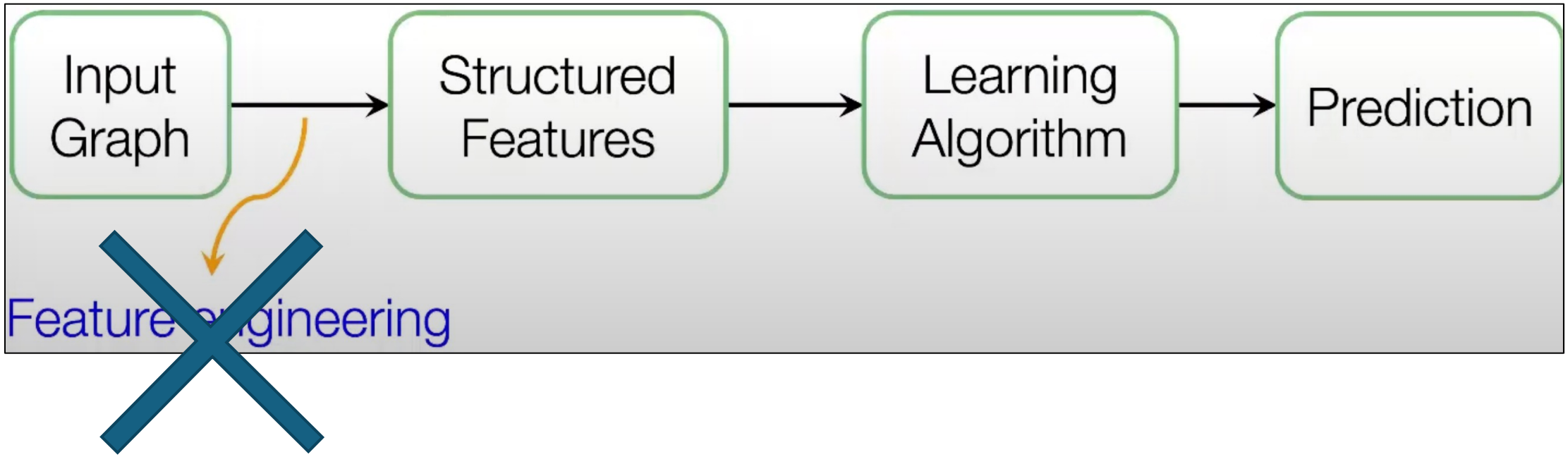
Node features

	Bags of nodes	...	Generalized paths	...	
1			X		
2			X		
3	X		X		
4	X				



Source : Jure Leskovec, CS224W, Stanford

From feature engineering to representation learning



Source : Jure Leskovec, CS224W, Stanford

Principe de l'apprentissage de représentation

- Pour des parties du graphe (les nœuds), un **encodeur** produit une représentation sous forme de vecteur numérique = *Embedding* ou plongement

- Le **décodeur** associe un score à l'embedding

- La fonction de perte compare (valeur prédite et valeur réelle) + la descente de gradient oriente les mises à jour

Random walks & node2vec

Random walks & node2vec

- Idea: if random walk starting from a node u visits v with a high probability, u and v are similar
- Similar nodes are nearby in the embedding space
- Finding node u embedding $\mathbf{Z}_u$ that minimize $\mathcal{L}$:

$$\mathcal{L} = \sum_{u \in V} \sum_{v \in N_R(u)} -\log\left(\frac{\exp(\mathbf{z}_u^{\mathrm{T}}\mathbf{z}_v)}{\sum_{n \in V}\exp(\mathbf{z}_u^{\mathrm{T}}\mathbf{z}_n)}\right)$$

sum over all nodes u

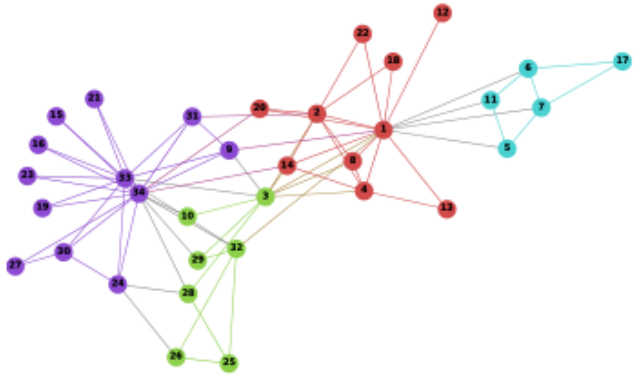
sum over nodes v seen on random walks starting from u

or $P(v|\mathbf{Z}_u)$

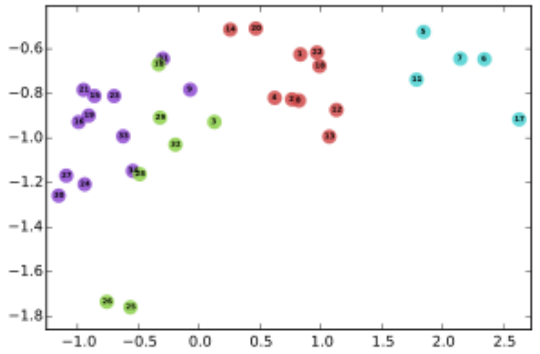
predicted probability of u and v co-occuring on random walk

i.e., a softmax functio

DeepWalk



(a) Input: Karate Graph



(b) Output: Representation

DeepWalk, Perozzi, Al-Rfou & Skiena, KDD 2014

+SkipGram

>negative sampling

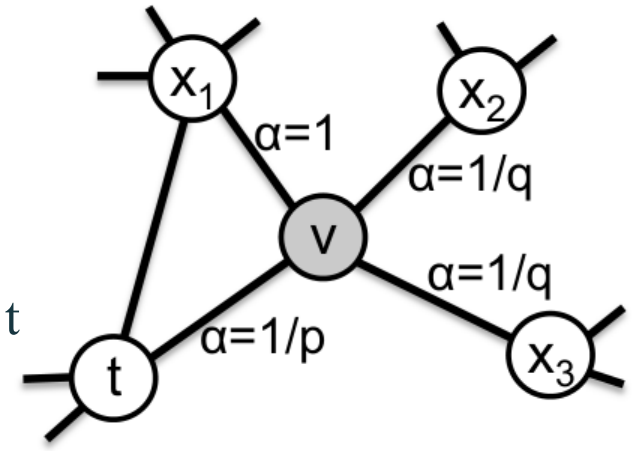
$$\mathcal{L} = -\log \sigma(u_c^\top z_v) - \sum_{n \in \mathcal{N}^-} \log \sigma(-u_n^\top z_v)$$

Contexte observé
dans les marches aléatoires

Contexte tiré alétoirement dans
un ensemble aléatoire $\mathcal{N}$

Node2vec : Contrôle de l'exploration aléatoire avec p et q

v : nœud courant
t : nœud précédent
x_i : nœuds candidats
X_1 est aussi voisin de t



$$P(x \mid v, t) \propto w_{vx}\, \alpha_{pq}(t, x)$$

Candidat	Distance à t	Poids	Probabilité
t	0	1/p = 0,5	1/11
x_1	1	1	2/11
x_2	2	1/q = 2	4/11
x_3	2	1/q = 2	4/11

Exemple : p = 2 et q = 0,5

p élevé : évite les retours en arrière
q élevé : exploration plus locale
q faible : davantage de sorties vers des nœuds non voisins des précédents

Arêtes de même poids. Normalisation : 0,5 + 1 + 2 + 2 = 5,5.

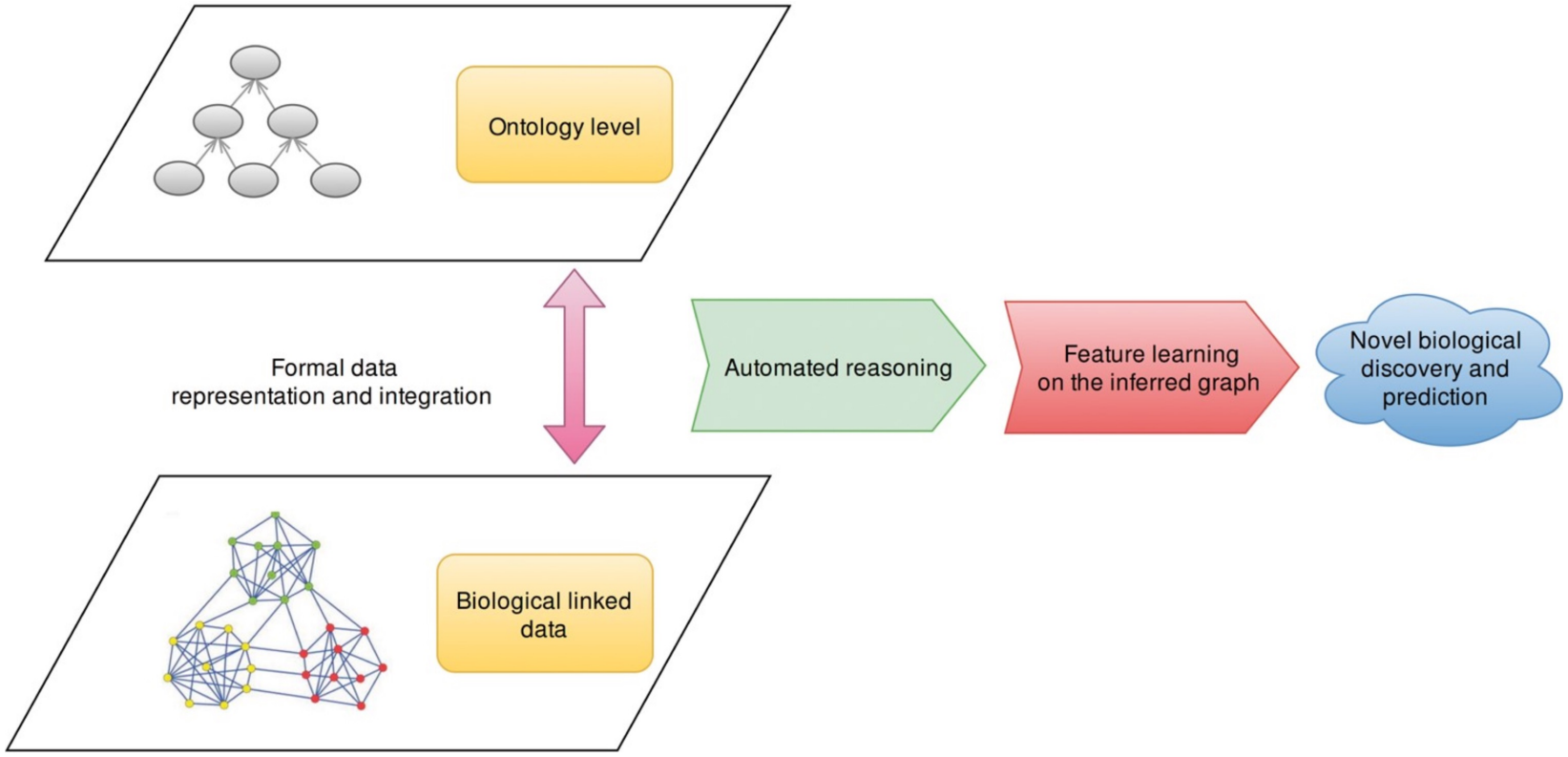
Grover & Leskovec, KDD 2016

RDF2vec : inclut les prédicats RDF dans les marches aléatoires

sujet	prédicat	Objet
drug_A	causes	disease_1
drug_B	treats	disease_1
...		

> Considère les prédicats dans les marches
> Permet de distinguer les paires liées par des prédicats différents
> Adaptation à un modèle Word2vec
[Ristoski & Paulheim, ISWC 2016]

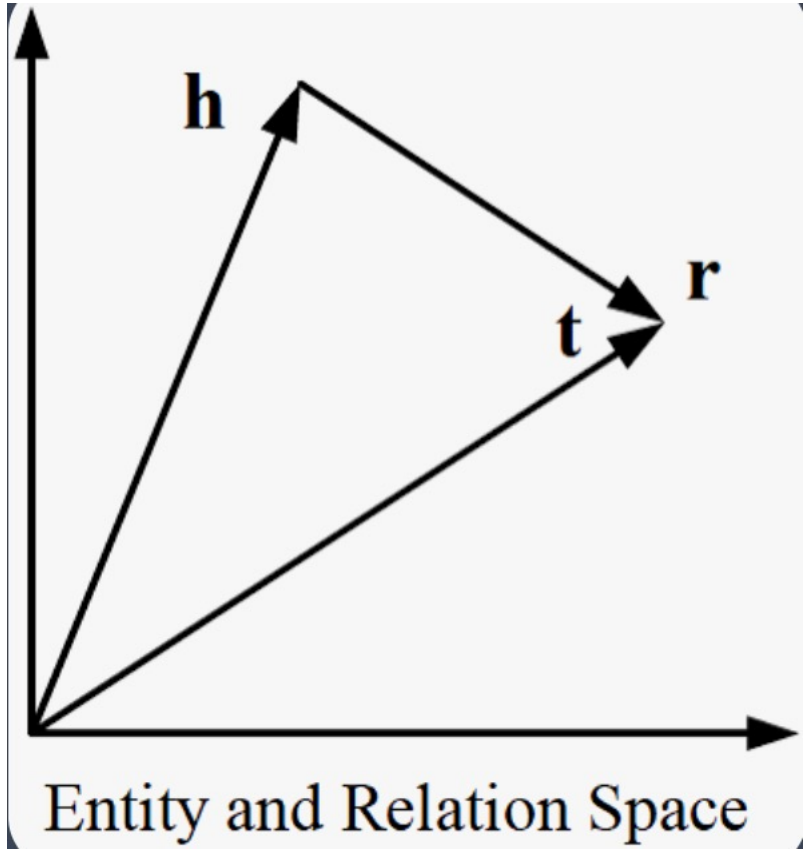
Marches aléatoires et connaissances biologiques intégrées
Alshahrani et al., *Bioinformatics* 2017



1. Extension du graphe par raisonnement
2. Prédiction de : fonctions de protéine, gène-maladie, protéine-protéine et protéine-médicaments
3. L'extension améliore parfois la prédiction, parfois dégrade, parfois n'a pas d'impacte

Approches Translationnelles *(TransE-like)*

TransE



$h+r \approx t$

objectif : apprendre des embeddings pour les entités et les relations tels que

$$d(h,r,t) = ||z_h + z_r - z_t||_1$$

Source: dglke

$$\mathcal{L} = [\gamma + d(h,r,t) - d(h,r,t^-)]_+ \qquad [x]_+ = \max(0,x)$$

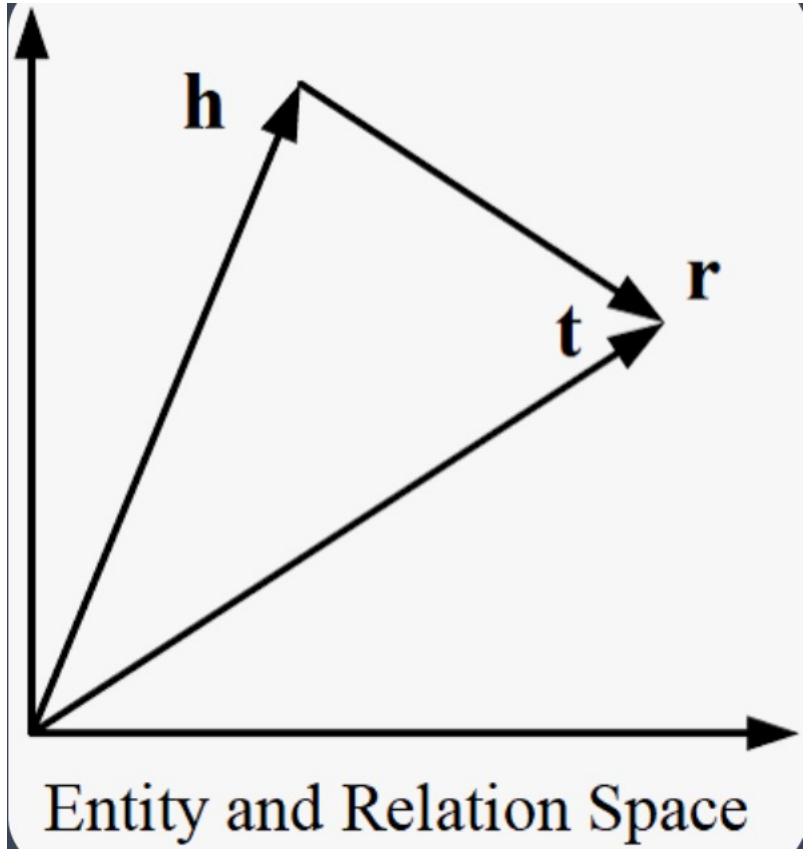
marge

Tripet observé (exemple positif)

Triplet fictive corrompu (exemple négatif)

[Bordes et al., NeurIPS 2013]

TransE



$h+r \approx t$

objectif : apprendre des embeddings pour les entités et les relations tels que

$$d(h,r,t) = ||z_h + z_r - z_t||_1$$

Source: dglke

$$\mathcal{L} = [\gamma + d(h,r,t) - d(h,r,t^-)]_+ \qquad [x]_+ = \max(0,x)$$

marge

Tripet observé (exemple positif)

Triplet fictive corrompu (exemple négatif)

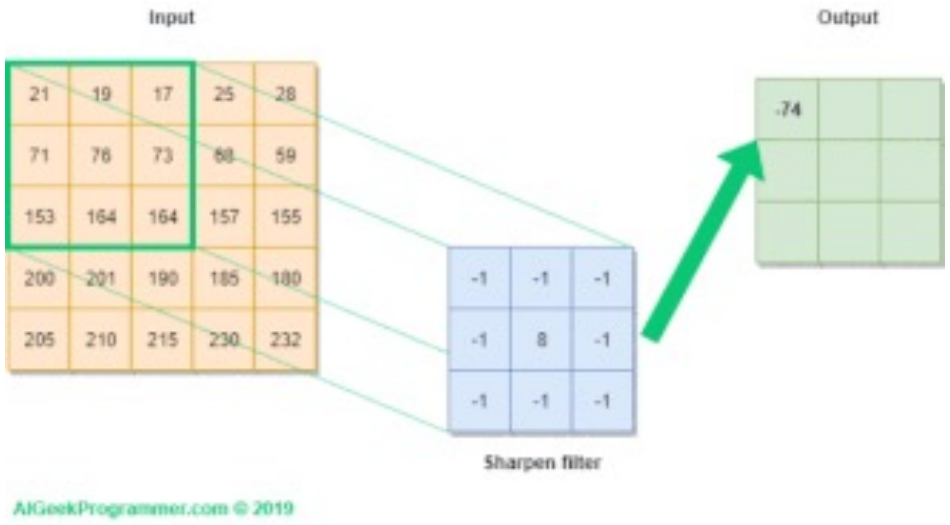
>Les triplets ficitfs, peuvent être valides (même si absents du KG)

[Bordes et al., NeurIPS 2013]

Après TransE

TransH *: projection dans un hyperplane…*
TransR *: …ou dans l'espace des relations*
TransD : projection dynamique
StransE : deux matrices par relation
ToruseE : translationsur un tore
RotateE: rotation à la place de la translation
…

Graph Convolutional Networks



Principe de CNN

pour les images, un petit filtre agrége les valeur d'un voisinage local

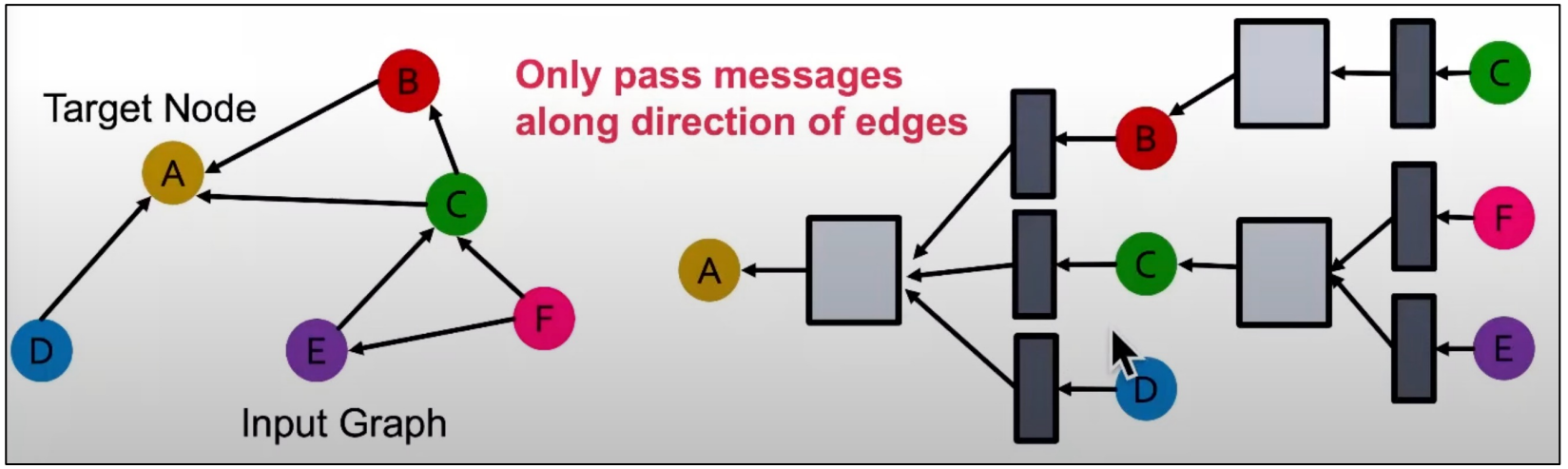
cela rend possible la detection de motifs répétés

on apprends les paramètres du filtre

il s'agit de passer une agrégation spaciale dans l'image à une agrégation adaptée à la topologie des graphes

Message passing framework

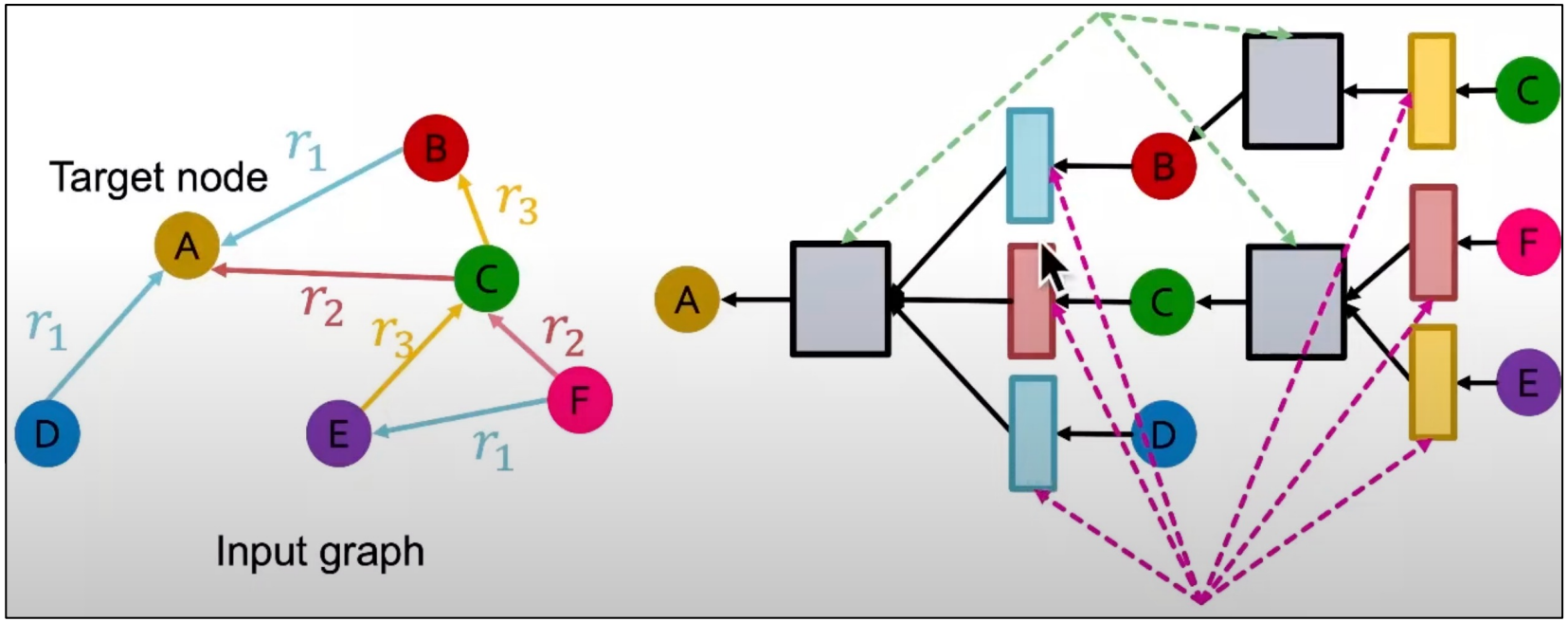
- intuition



Source : Jure Leskovec, CS224W, Stanford

Message passing framework

- Intuition, with various types of relations



Source : Jure Leskovec, CS224W, Stanford

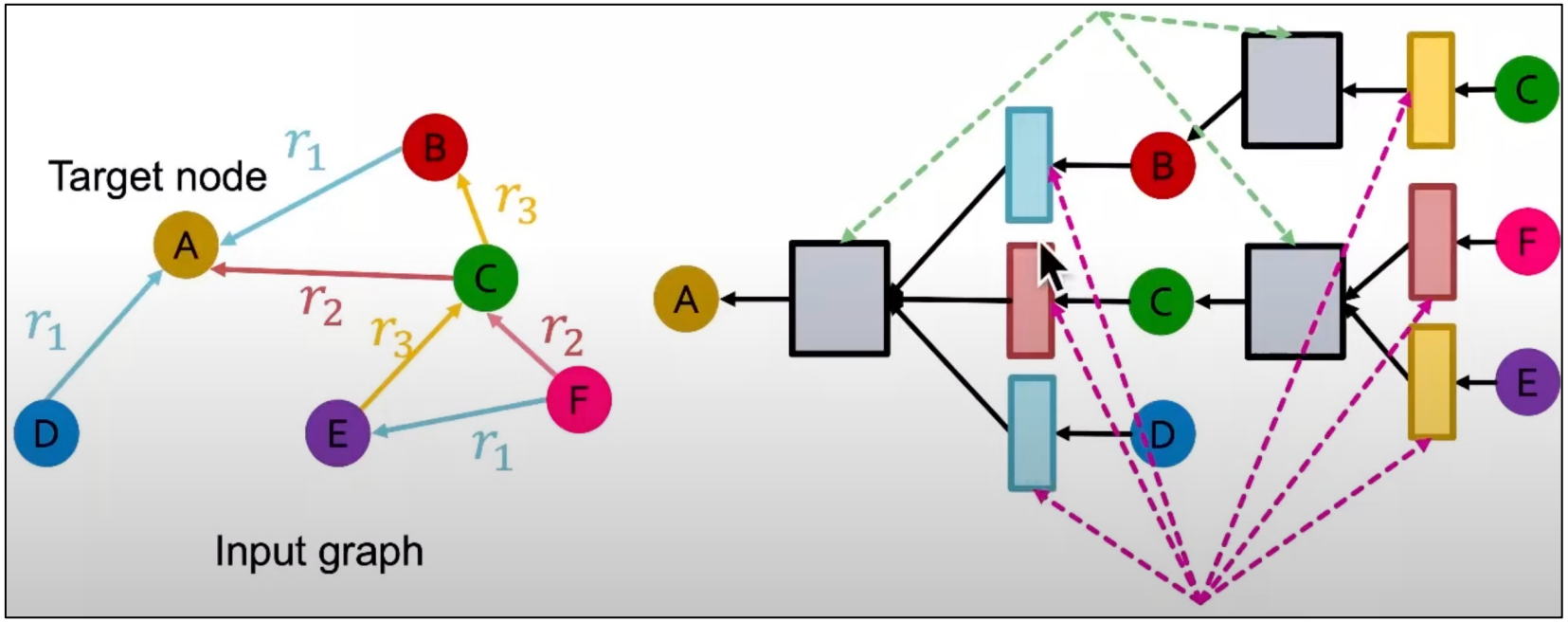
Message passing framework

- Intuition,

with various

types of relations

- Kipf and Welling, 2017



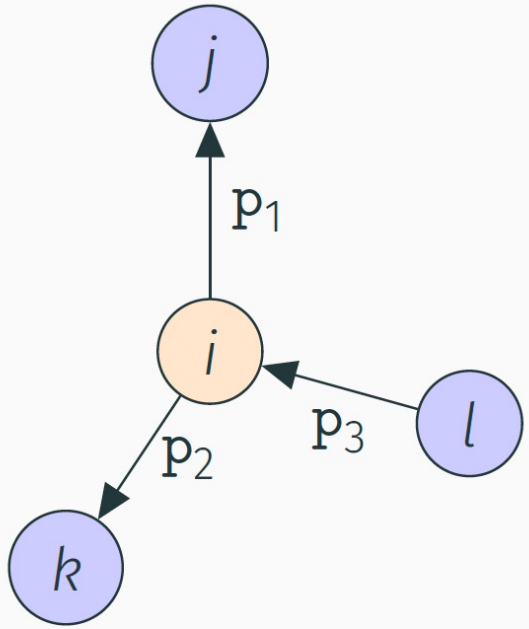
$$h_i^{(l+1)} = \sigma\left(\underbrace{\sum_{r\in\mathcal{R}}\sum_{j\in\mathcal{N}_i^r}\frac{1}{c_{i,r}}W_r^{(l)}h_j^{(l)}}_{\text{Neighborhood}} + \underbrace{W_0^{(l)}h_i^{(l)}}_{\text{Self-connection}}\right)$$

Source : Jure Leskovec, CS224W, Stanford

Learning node embeddings with Graph Convolutional Networks (GCNs)

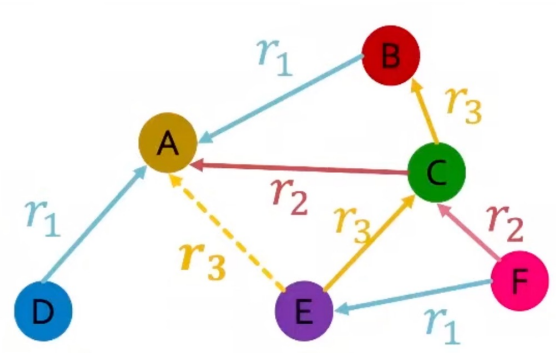
- "Message-passing framework" of multiple layers (Kipf et al. 2017; Schlichtkrull et al. 2018)
- Learns embeddings of a node *i* w.r.t. its neighbors

$$h_i^{(l+1)} = \sigma\left(\underbrace{\sum_{r \in \mathcal{R}} \sum_{j \in \mathcal{N}_i^r} \frac{1}{c_{i,r}} W_r^{(l)} h_j^{(l)}}_{\text{Neighborhood}} + \underbrace{W_0^{(l)} h_i^{(l)}}_{\text{Self-connection}} \right)$$



Source : Pierre Monnin

Link prediction with GCN



- (E, r_3, A) ?
- Use GCN to score (E, r_3, A):
  - Final layers for A and E provides their embeddings h_A^L, h_E^L
  - With a relation specific function:

$$f_r(h_E, h_A) = h_E^T W_r h_A^L$$

  - Define a set of positive and of negative (always a tradeoff!)
  - Define a loss, *e.g.*, cross entropy

- Other tasks from graph embeddings: node **classification**, **clustering, ...**

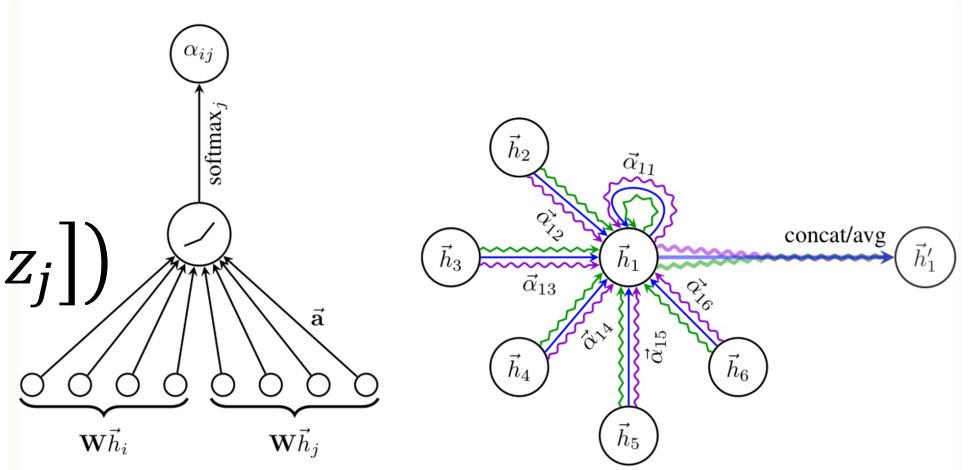
Graph Attention Networks

GAT (Graph Attention Network)

- Aggrège les representation des noeuds voisins avec des poids appris qui peuvent différer d'un voisin à l'autre : l'attention

$$=> \quad z_i = W h_i$$

$$e_{ij} = \text{LeakyReLU}\left(a^\top [z_i \| z_j]\right)$$



$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k \in \mathcal{N}(i)} \exp(e_{ik})}$$

- Les poids s'apprennent avec la tâche finale, par rétropropagation
- Plusieurs têtes d'attention : initialisation différente = combinaison de poids différents
- Plusieurs couches = augmentation de la taille du voisinage (comme pour les GCN)

HGT (Heterogeneous Graph Transformer)

- D’autres méthodes d’attention considèrent le type des nœuds et le prédicat :
  - HGT : Hu, Dong, Wang & Sun. Heterogeneous Graph Transformer. WWW 2020. doi:10.1145/3366423.3380027
  - RED-GNN
  - ULTRA

En pratique

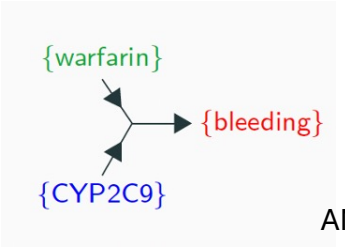
- 2 exemple de librairies :
  - pyKeen : https://github.com/pykeen/pykeen

  “Python KnowlEdge EmbeddiNgs is a Python package designed to train and evaluate knowledge graph embedding models (incorporating multi-modal information).”
  - PyG : https://pytorch-geometric.readthedocs.io/

  “is a library built upon PyTorch to easily write and train Graph Neural Networks (GNNs) for a wide range of applications related to structured data.”

Exemples d'applications

Identification de pharmacogènes
Dalleau K, et al. 2017



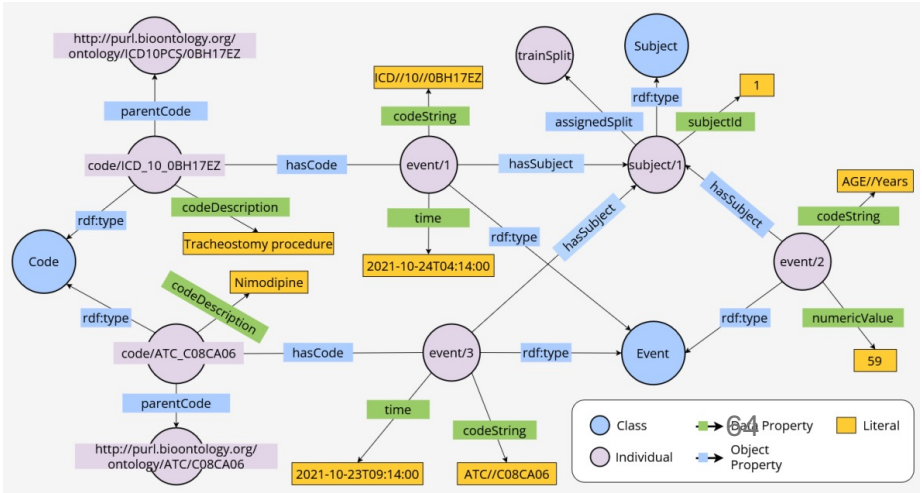
ANR PractiKPharma 2016-21

Résolution d'entité dans un graphe hétérogènes de connaissances biomédicales
Monnin P, et al. 2021

Explication des mécanismes des effets secondaires
Bresso E, et al, 2021

Rapprochement de protocoles de traitements similaires
Jhee JH et al, 2024

Prédiction d'évolution clinique
Jhee JH, et al, 2025



MAKAAO
Index of human autoantibodies

Comparaison des connaissances sur les maladies autoimmunes
Maury F, et al, 2026

...

PEPR NEUROVASC 2023-27

Conclusion

- 2 façons d'acquérir des connaissances
  - Construction d'ontologies et Raisonnement symbolique
  - Apprentissage supervisé à partir de KG



- Question de recherche :

  Comment combiner les deux ?

Des pistes à partir d'embeddings de graphes de connaissances (TransE, MulE, *etc.*)

où *embedding* et fonction de perte son adaptées en fonction de la sémantique associée aux types de nœuds du graphe, *e.g.*, objet *vs*. Classe d'une hiérarchie ou objet *vs*. valeur

ex: transformations hyperboliques pour considérer les distances dans les hiérarchies

Merci pour votre attention

Avez-vous des questions ?

Contact: adrien.coulet@inria.fr

Principales references

- Lisa Ehrlinger and Wolfram Wöß. “Towards a Definition of Knowledge Graphs”. In: Joint Proceedings of the Posters and Demos Track of the 12th International Conference on Semantic Systems - Vol. 1695. CEUR Workshop Proceedings. CEUR-WS.org, 2016. url: http://ceur-ws.org/Vol-1695/paper4.pdf.
- Heiko Paulheim. “Knowledge graph refinement: A survey of approaches and evaluation methods”. In: Semantic Web 8.3 (2017), pp. 489–508. doi: 10.3233/SW- 160218. url:https://doi.org/10.3233/SW-160218.
- Aidan Hogan *et al*. 2021. Knowledge Graphs. ACM Comput. Surv. 54, 4, Article 71 (May 2022), 37 pages. https://doi.org/10.1145/3447772.
- Gysi DM, et al. Network Medicine Framework for Identifying Drug Repurposing Opportunities for COVID-19. ArXiv [Preprint]. 2020 Apr 15:arXiv:2004.07229v2. Update in: Proc Natl Acad Sci U S A. 2021 https://pubmed.ncbi.nlm.nih.gov/32550253/
- Wang S, et al. Leveraging the Cell Ontology to classify unseen cell types. Nat Commun. 2021 Sep 21;12(1):5556. doi: 10.1038/s41467-021-25725-x. PMID: 34548483 https://pubmed.ncbi.nlm.nih.gov/34548483/
- Ruiz C, et al. Identification of disease treatment mechanisms through the multiscale interactome. Nat Commun. 2021 Mar 19;12(1):1796. doi: 10.1038/s41467-021-21770-8. PMID: 33741907
- Zitnik M, et al. Evolution of resilience in protein interactomes across the tree of life. Proc Natl Acad Sci U S A. 2019 Mar 5;116(10):4426-4433. doi: 10.1073/pnas.1818013116. Epub 2019 Feb 14. PMID: 30765515
- Bresso E, et al.. Investigating ADR mechanisms with Explainable AI: a feasibility study with knowledge graph mining. BMC Med Inform Decis Mak. 2021 May 26;21(1):171. doi: 10.1186/s12911-021-01518-6. PMID: 34039343
- Dalleau K, et al. Learning from biomedical linked data to suggest valid pharmacogenes. J Biomed Semantics. 2017 Apr 20;8(1):16. doi: 10.1186/s13326-017-0125-1. PMID: 28427468;
- Thomas N. Kipf and Max Welling (2017). “Semi-Supervised Classification with Graph Convolutional Networks”. In: ICLR 2017, https://openreview.net/forum?id=SJU4ayYgl.