Comptes rendus d'article : quelques notes

Article 1 : Un nouveau modèle pour l'extraction d'informations : ReLiK: Retrieve and LinK, Fast and Accurate Entity Linking and Relation Extraction on an Academic Budget

https://arxiv.org/pdf/2408.00103

Papier typique qui propose un nouveau système pour réaliser une tâche de NLP.
Souvent ces papiers se structurent en trois parties
1) présentation de la littérature et de l'état de l'art sur la tâche concernée
2) présentation de l'architecture / du modèle
3) évaluation pour démontrer qu'il dépasse l'état de l'art
(* éventuellement critiques et perspectives d'améliorations)

Objectif du papier :
=> présenter ReLiK (Retriver Reader Architecture) qui doit réaliser les tâches d'Entity Linking et d'extraction de relations : préciser que l'on parle de **closed IE** par rapport à une liste de relations possibles pré déterminées..

1) literature review (étendue entre l'intro et les parties 2 et 3 :
basée sur trois « propriétés fondamentales » : vitesse d'inférence, flexibilité et performance.
qui ne sont pas précisément définies. Argument rhétorique pour se démarquer d'autres systèmes.
Approche basée sur l'architecture Retriever-Reader issue d'un papier de 2017 (Chen et al.).
j'attends du groupe qu'il revienne sur cette architecture et la présente plus en détail.
Lien avec l'Open Domain Question Answering en deux étapes.
Architecture basée en deux temps :
=> Retriever : prend en entrée un texte et renvoie les possibles entités et relations
=> Reader : prend en entrée le texte et la sortie du Retriever et doit refaire le lien avec le texte original (span)

Comment les 3 objectifs sont ils atteints ?
Vitesse => avec une approche de retrieve, plus besoin de modèle à gros paramètre et un modèle plus léger suffit
flexibilité => l'utilisation d'un retriever permet une adaptation plus facile à d'autres domaines.
Performance => le travail de feature engineering et le fait de traiter les deux tâches à la fois) améliore les résultats.

Critique:
la literature review n'est pas très organique, séparation entre la définition des tâches et la manière de les traiter. Ça manque de liant pour justifier toute la démarche.
Manque la définition de ce qu'est une relation pour le système et des exemples incorporés dans le coeur du papier. (autrement que dans le schéma)
Dans la définition d'extraction de relations, manque de clarté au départ sur le fait qu'on parlait d'extraction de relations **dont la liste a été définie par avance**. On est donc pas ici sur de la OIE.
Manque de clarté sur ce qu'on Retrieve avec le retriever. Comment est-il initialisé ?
Il faut arriver à un certain point de l'article pour le comprendre.

2) présentation du système

le retriever : basé sur des approches de recherches documentaire.
Entraîné avec NCE. (à définir)
Reader :
modèle de type BERT. Avant <SEP> texte, après la liste des entités et relations trouvées par le retriever. Ajout de nouveaux tokens spéciaux. Encodeur
fait un single forward pass et encore le texte orginal avec des balises Begin, inside
modèle entraîné à prédire pour chaque token (dans la représentation vectorielle) d'être le début ou la fin d'une mention. On ne précise pas exactement quel est le format de sortie.
Entity linking : nécessité de faire le lien entre les mentions repérées par le Reader et les entités et relations trouvées dans le retriever.
Pour cela comparaison entre la représentation vectorielle de la mention et celle de l'entité.

$$\mathcal{L} = \mathcal{L}_S + \mathcal{L}_E + \mathcal{L}_{EL}.$$

Les deux tâches de read start, read end (deux tâches de classif) et Entity linking sont traitées en même temps. Forme de joint training même si le terme n'apparaît pas.

Extraction de relations : à partir de toutes les relations trouvées, on fait une représentation de chaque triplet avec le produit de Hadamard (à définir)

$$T_{m,m',k} = S_m \odot O_{m'} \odot R_k \in \mathbb{R}^H$$

et on s'en sert pour voir si
le texte la contient.

Si je comprend bien l'article il y a deux versions du modèle une pour EL une pour RE. La partie purement « reader » qui marque le span des entités est commune mathématiquement.
Possibilité d'entraîner une version qui fait les deux d'un coup.

critique : un papier qui est difficile à lire pour quelqu'un qui ne travaille pas exactement sur les mêmes problématiques.

3) évaluation du système

Fait distinctement pour les deux tâches
définir micro F1

* EL
entraînement du modèle sur un dataset et évaluation sur un autre.
S'inspire d'un papier de Zhang (2022) qui n'est pas présenté.
La mention de cet article aurait dû apparaître plus tôt dans la review.
Les progrès sont marginaux par rapport à ce dernier article.
Le problème des entités qui ont plusieurs formes textuelles dans les textes n'est que trop rapidement évoqué et esquivé par la mention de dictionnaire remplis manuellement.
* R E
seulement 24 types de relations / 5 dans les corpus testés. Pas de visibilité au-delà.

Là encore les gains sont marginaux par rapport au précédent SoTA, seul le temps d'inférence est amélioré.

4) critique :
reconnaissance du besoin de tester sur d'autres domaines et textes plus variés.
Manque une étude d'ablation sur le rôle de chaque composant sur les autres.
Pas de discussion sur la notion de topK ni sur le moyen de choisir le K
Manque une étude sur l'influence positive qu'aurait un joint training pour RE et EL (on s attend à ce que le second aide le premier, du moins ne dégrade pas)

Critique globale :
article qui reprend une nouvelle architecture pour traiter à la fois ER et EL avec une phase de retrieve sur une base de données existante.
Mais article trop dépendant de celui qui a proposé l'architecture.
Les auteurs par leurs tests parviennent à prouver que les trois objectifs sont tenus par les choix d'architecture.

Ne prend pas le temps de définir tous les concepts. Beaucoup de jargon.
Papier mal organisé (le nom de certaines parties et de certaines sous parties est le même)
Manque total de transition entre les parties pour guider le lecteur.
Le papier ne précise pas assez clairement qu'il propose une architecture. La mention explicite des modèles fine tunés n'arrive qu'à la fin.
Manque de précision sur le type de données devant être fournies au retriever. Nombre de relations ? D'entités ? Tests pour voir si les performances se dégradent quand grandit la taille de cette base de données. La phase de retrieve n'est pas assez évaluée en soi, on a juste une évaluation extrinsèque de tout le système. Est il viable de faire un recall 100 ? c'est beaucoup de documents extraits.

Article 2 : survey sur l’ embedding d’ontologies

https://arxiv.org/pdf/2406.10964

Cet article est un survey.
Type d’articles qui fait le tour d’une question scientifique pour faire le point et faciliter l accès à l information. Le but est de centraliser les connaissances sur un sujet.Le sujet doit être assez ouvert pour que la synthèse soit utile et pouvoir recouper assez d’articles, mais pas trop non plus sinon on en reste à des généralités.
Un survey s’inscrit aussi sur la durée.
Ce qui se passe est que sur un sujet à la mode, souvent plusieurs survey sont écrits.
Les survey se construisent et se répondent les uns sur les autres soit en explorant des sous thèmes différents, soit en opposition en venant combler le manque d’un survey précédent.
Un bon survey est clair, exhaustif et bien structuré

introduction :
claire présentation des ontologies
bon background sur les embeddings mais manque de liant, les deux sujets auraient pu être séparés
un peu rapide sur Trans E mérite un schéma
ne précise pas clairement la difficulté
se différencie des embeddings de KG qui cherchent à construire des embeddings de la relation

ici survey mais pas que : la partie 3 présente un nouveau système d embedding d ontologies

II) background
la distinction ontologie KG est elle claire ?

Présentation utile pour les personnes venant d’un autre background.
Mais manque de recul sur la proximité de toutes les définitions et le fait que l’on a affaire à un continuum !
J attends une présentation claire de ce que c est que Abox Rbox Tbox

Section sur les embeddings légère pour février 2025…
le point clarifiant la def d embeddings aurait dû arriver bien plus tôt selon moi. Début de partie voire intro.

N expliquent pas clairement et explicitement ce qui fait que l embedding d une ontologie est différent de celui d’un KG. La différence est qu’une ontologie capture des règles de raisonnement et d’inférences (les règles dont on a pas eu bcp le temps de parler).

Les stratégies d embedding
Survey organisé par complexité des ontologies pas par méthode technique. Dommage
 manque une illustration des concepts géométriques. J en attends une de vous ;

ontologies with litterals = complex graphs
Il aurait fallu des liens avec les travaux sur les embeddings de graphes compexes.

La partie IV manque une intro / transition

partie un peu courte sur les emplois en KE (manque l ajout de nouveaux termes à l ontologie ?)
et tres decevante sur le reste.
2025 ! rôle des LLM ? Connexion aux bases de données ?

V other applications ; très léger !

VI : pub pour MOWL : tres abstrait.
Ne justifie pas la dépendance au format OWL !

VI défis :
la partie sur les LLMs est délirante

Critique :
réduit les ontologies à certains de leur formalismes (notamment RDF et OWL) et part de là et pas de la définition globale d'une ontologie.
C'est un faux survey « smuggle » un travail d'une librairie, ce qui est hautement inhabituel !
Les différentes approches d'embedding ne sont pas à mon sens suffisamment explicitées !
Le lien avec les travaux sur l embedding de graphes complexes ne sont pas faits or il y a plus de monde sur la question et des avancées.

Organisation correcte même si discutable, plan tenu, mais l impression de survol.

Article 3 : Talk like a graph : encoding graphs for LLM

Question centrale : Que signifie « raisonner sur un graphe » ?

test des capacités de raisonnement des LLMs sur les graphes.
Essaie de démontrer que 3 paramètres entraînent de la variation :
méthode d'encodage
nature du graphe
structure du graphe (topologie)

On s attend donc à l'application de la démarche scientifique, c'est à dire ne faire varier qu'un paramètre et voir ce qui va changer.

Question : peut-on envisager d'autres facteurs ? Le prompt ?

Séparation du prompt et de l'encodage du graphe.
Point important on considère le LLM comme figé, on ne va pas toucher à ses paramètres.

Quelles sont les tâches qui ont été testées ? Que pouvez-vous dire ?
Elles sont basiques !

Limitations de Graph QA ? Liens avec de véritables cas d'usage ?
N'est ce pas mieux de faire du python facile pour traiter ce genre de choses ?

Comment pourrait-on améliorer la compréhension des graphes par les LLMs ?

Présenter les différents types d'encodage retenus. GoT ?are you serious ?
Sans surprise les notations mathématiques les meilleures.
Liste des edges ?

Topologie : l idée est extrêmement intéressante
« For example, the cycle check task achieves 91.7% accuracy on complete graphs and 5.9% accuracy on path graph » => un peu évident ...

aussi tester la taille du graphe

annexe : la partie la plus claire du papier

Choix du LLM ils ont joué corporate mais ça aurait été intéressant de tester gpt4

interrogation sur la diversité d'un graphe.
« graph generator ? Comment ça fonctionne ?
Peut on penser à une autre forme de diversité ?

ccl : série de petites expériences intéressantes . Article rafraîchissant mais nécessité de pousser plus loin.
Des constats pertinents mais manque de questionnement sur le pourquoi ou d'études pour justifier la variabilité observée.

Le r de strawberry !!! le problème vient de la tokenization.

Article 4 : From Local to Global: A Graph RAG Approach to Query-Focused Summarization

https://arxiv.org/html/2404.16130v1

Aussi appelé Microsoft graph RAG

Pbtique : le RAG ne parvient pas à répondre à des questions générales sur la base
proposition d'une approche pour y parvenir. Query Focused Sumarization.

Plan
1) état de l'art
limitations du RAG :
sensemaking questions selon le volume des données.
Différence avec autres approches graph RAG. Notion de modularité

2) description du process et du protocole d évaluation
Processus en 2 étapes
construction d'un KG par LLM
partition selon une hiérarchie de communautés.
Variété de l'algo de louvain
Génération de résumés de communautés.
Quand l utilisateur pose une question, approche map reduce des résumés
evaluation par LLM as a judge avec dataset synthétique

lien avec la tâche du résumé abstractif

3) expériences
1 Million token = 250k mots pas si grand, un gros roman.
* Comparaison de diverses implémentations avec un RAG classique
* analyse des « claims » ?? montrer que la réponse est plus complete.
PK c'est évident ?

4) limitations
généralization a d autres corpus et taches
utilité de l hybride.
Section vraiment légère.

* réflexions sur le coût ?
* réflexions sur la perte d'information ?
* pb dans l évaluation : risque que les questions ne correspondent pas au corpus.
* critères d évaluation : pourquoi s éloigner de ceux du RAG classique ? Empowerment ? Pardon ?
* 8k fenêtre de contexte : très court selon les standards actuels (128) mais faisait sens à l époque ;
* pb des critères d évaluation qui sont un peu autoréalisateurs. On a décrit des critères pour que le RAG échoue

Questions :
pourquoi utiliser 2 llms différents pour LLM as a judge
Question : c est quoi la modularité ?
Comment appelle t on la méthode consistant à un LLM de juger l output d'un système ?
En quoi cette approche reste-t-elle du RAG ?

Article 5 : A-M EM: Agentic Memory for LLM Agents

https://arxiv.org/abs/2502.12110

proposition d'un système de mémoire agentique : le LLM peut dynamiquement mettre à jour sa mémoire. « dynamic indexing and linking »

1) intro : constat que les LLMs ont besoin de mémoire et que les systèmes actuels sont basiques. Index and retrieve cf RAG.
Défaut : côté strict
question de recherche :
« Therefore, how to design a flexible and universal memory system that supports LLM agents' long-term interactions »

emploi de Zettelkasten method : mode de gestion de l'information comme notes atomiques qui peuvent être reliées.

Agentique : quand un nouvel élément de mémoire est ajouté, le système analyse les notes précédentes. Évolution dynamique des faits.
2 opérations, link generation,
memory evolution

2) état de l'art.
Un peu répétitif de l'intro
autres mécanismes : indexation (dense retrieval) ou read_write structure
réf de MemGPT cache qui priorise le récent.
Mais critique que ce sont des workflows fixes. => mauvaise généralisation selon le cas d'usage.
Inspiration des systèmes de RAG agentiques qui décident quoi et quand recherche.

3) méthodologie
3 opérations
note construction.
Une note est un objet contenant beaucoup d'éléments, le texte de l'interaction, des tags de catégorisation… analysé par LLM. On en fait un embedding. autonomous extraction of implicit knowledge from raw interactions

link generation : par similarité sémantique. Top k puis analyse par un LLM pour valider le lien. Risque de pertes ici !

Memory evolution.
Top k éventuellement mis à jour selon le contexte de la nouvelle note.

Le retrieval se fait par similarité vectorielle.

4) expériences

difficulté de trouver un dataset adapté : longues conversations.
Mais donne t on le contexte ? 9K tokens trivial aujourd hui

critiques :
comment les liens sont-ils exploités ?
D'où vient leur estimation chiffrée du nombre de tokens utilisés ?
Reproductibilité des résultats ?
Réflexion sur les limitations bien trop légère. On ne parle absolument pas de coût …
Manque aussi de réflexion sur l'importance de la taille du modèle et de la taille de la fenêtre de contexte sur le résultat final.

Question : combien d'appels de LLM pour le stockage d'une note ? = 1 tour de parole
que pensez-vous des métriques d'évaluation choisies ?
Au final , que constitue l'ensemble des notes ?
Pourquoi des risques d'hallucinations ?
what is Ollama ?