Générer les règles : distillation symbolique d'un LLM pour une extraction d'information frugale au service du renseignement

Mickaëla Mastrodicasa, Louis Jourdain — ChapsVision

Identification des cas d’usage

Les services de renseignement ont besoin d'agréger les informations issues de sources hétérogènes pour reconstruire des connexions pertinentes qui ne sont pas perceptibles à l'échelle du document unique. Pour l'analyse de corpus massifs, les graphes de connaissance sont un outil précieux : leur topologie permet d'identifier des acteurs clés ou d'étudier les liens entre plusieurs entités, et on les exploite comme source d'information pour les modèles d'IA (Graph RAG) comme pour la prise de décision. Si leur exploitation est intuitive et facile, c'est la construction de graphes de qualité, adaptés au cas d'usage, qui a freiné leur plus large adoption. Les LLM ont donné l'illusion d'une construction facile, mais les graphes ainsi produits avec enthousiasme sont rarement évalués, et les approches par prompting se dégradent dès qu'on quitte les petits corpus bien délimités [2]. Le mur est aussi économique. Notre corpus d'étude, UkrainIA, rassemble 180 000 documents de presse francophone sur le conflit russo-ukrainien, dont sont issues environ 6 millions de relations candidates. Les faire trancher une à une par un LLM propriétaire reviendrait à 0,07 $ par décision, soit de l'ordre de 420 000 $ pour traiter le corpus une seule fois, au tarif catalogue et sans mise en cache. Or ce tri est indispensable : notre annotation de référence montre que 70 % de ces relations sont inexploitables en l'état. Ainsi (Emmanuel Macron / s'est rendu à / Washington) est exploitable, quand (En France / Moscou peut compter sur le soutien de / Stars des réseaux sociaux) ne l'est pas : « En France » est un complément de lieu en tête de phrase, non l'agent du verbe, dont le sujet réel est « Moscou ». S'y ajoute la souveraineté : ces corpus ne peuvent transiter par des API externes, et l'extraction doit être réalisée par des composants légers tournant on premise, voire sur machine air-gappée. Les utilisateurs visés sont les analystes du renseignement multi-sources. Les mêmes contraintes se retrouvent en lutte anti-fraude et en investigation financière. Ce travail prolonge la démarche engagée à CAID 2025 sur la canonicalisation frugale des entités du même corpus [1], en la portant aux relations, dont la variabilité de forme rend le problème sensiblement plus difficile.

Approche technique proposée

Générer puis filtrer : plutôt que de confier l'extraction des relations à un modèle génératif, nous la traitons comme une génération combinatoire de candidats suivie d'un filtrage appris : toute paire d'entités d'intérêt repérée dans la même phrase constitue une relation candidate, et un classifieur décide ensuite si ces deux entités sont effectivement reliées par une relation sémantiquement pertinente, puis en extrait le cœur. Le rappel devient ainsi une propriété du générateur, garantie par construction, et le filtre ne porte que la précision. Cette combinatoire par phrase est précisément ce qui produit les 6 millions de candidats du corpus : tout l'enjeu se déplace vers un vrai tri de ce vivier, et le paradigme n'est praticable qu'à la condition d'un filtre à coût marginal nul.

Les trois familles historiques de techniques d'extraction d'information imposent chacune un compromis, que nous avons quantifié. Les méthodes symboliques sont très rapides à exécuter mais demandent un travail humain considérable et doivent être reprises à chaque nouveau cas d'usage : sur le benchmark MINE [3], notre extracteur à base de règles seules atteignait 23,9 % de précision pour 6,0 % de rappel. Les modèles spécialisés [5] sont plus fiables mais réclament beaucoup de données d'entraînement et se transposent mal d'un domaine à l'autre. Les LLM sont les plus faciles à mettre en place et couvrent presque tout, 98,5 % de rappel, mais au prix d'une sur-extraction massive : 1 403 vrais positifs pour 6 281 faux positifs, chacun produit par un appel facturé, c'est-à-dire une dépense engagée pour produire du bruit. Peut-on obtenir le meilleur des trois mondes ? L'idée est de distiller les capacités d'extraction des LLM récents dans un système frugal : le modèle ne produit plus les relations, il produit les règles qui les trient. La littérature sur la distillation vise presque exclusivement le transfert d'un grand modèle vers un modèle élève plus petit ; nous visons le transfert vers un artefact symbolique. Cette voie contourne de surcroît un obstacle rarement

discuté : les conditions d'utilisation des LLM propriétaires interdisent généralement de réemployer leurs sorties pour entraîner un modèle concurrent, ce qui fragilise juridiquement toute distillation par apprentissage. C'est ce que nous avons nommé une **distillation symbolique**.

Étape 1 : Calibrer le LLM au niveau de l'expert. Le goulot d'étranglement est la donnée annotée, ce qui force à être précis sur la définition de ce qu'est un triplet valide. Nous employons un LLM comme annotateur [6], mais cela n'est viable que s'il annote comme le ferait un expert humain. Nous avons donc mis en place un protocole itératif où l'humain, assisté du LLM, écrit un guide d'annotation précis, structuré en trois vérifications syntaxiques : la source est-elle l'agent du verbe, le prédicat est-il isolable, la destination est-elle l'argument attendu. Négation, citation et modalité portent un flag propre et restent valides, de sorte que l'information de polarité n'est ni perdue ni confondue avec une erreur : point déterminant en analyse de renseignement. Le guide est raffiné par une boucle de modération [7], ensuite confiée à un agent qui l'édite et vérifie que la nouvelle version corrige les erreurs visées. Le résultat central est méthodologique : à modèle constant, la seule réécriture du guide fait passer le F1 de 0,544 à 0,804. La performance se joue dans la spécification, non dans le modèle. Un modèle ouvert hébergé on premise (Qwen3-32B) atteint alors un F1 de 0,726 pour une précision de 0,897, contre 0,699 pour un modèle propriétaire (Claude Sonnet 4.6) facturé 0,07 $ par annotation : rien ne justifie ici de payer l'inférence.

Étape 2 : Un moteur de règles auto-évolutif. Le LLM calibré ne sert plus qu'à générer assez d'exemples pour concevoir un système symbolique avoisinant ses performances. Nous nous sommes inspirés des travaux sur le *loop engineering* et des outils type Claude Code. Un agent ReAct [8] dispose d'outils pour lire et éditer un fichier de règles, simuler l'effet d'une modification avant de l'appliquer et tester les règles sur tout ou partie du corpus. Les phrases lui sont fournies avec la position des entités et leurs informations grammaticales (parties du discours, fonctions syntaxiques) [4]. Le moteur de règles fait office de juge : déterministe, extérieur à l'agent et non manipulable, il annule toute modification qui dégrade la métrique et renvoie le détail des erreurs introduites, ce qui est une forme de gradient textuel [9], qu'il utilise au tour de conversation suivant pour améliorer encore les règles. Doté d'une mémoire de ses tentatives passées, l'agent multiplie par 2,6 son gain par itération.

Une fois les règles apprises, le LLM n'est plus nécessaire et le composant peut être déployé sur machine air-gappée, sans GPU ni dépendance externe, déterministe, chaque décision traçable jusqu'à la règle qui l'a produite. L'humain est resté au centre du processus, analysant lui-même les données de terrain et s'assurant que le LLM produisait des résultats avec lesquels il est en accord. L'approche est adaptable à peu de frais à une nouvelle langue ou un nouveau corpus : quelques heures d'annotation suffisent à calibrer le LLM.

Partant d'un fichier de règles vide, l'agent construit un jeu de règles atteignant F1 = 0,981 sur le corpus de calibration, pour un budget d'inférence cumulé de 30 $, coût de dérivation du jeu de règles ; la consommation totale pendant le développement, est restée sous 1 000 $. Cette performance sera validée sur un corpus d'évaluation distinct, actuellement en cours d'annotation. Un démonstrateur est opérationnel sur UkrainIA, avec exploration du graphe de connaissance nettoyé et sera présenté lors de la conférence. Ce prototype, validé sur données réelles, est en voie d'industrialisation, avec extension des tests à d'autres corpus. La limite assumée porte sur le générateur de candidats : l'élargir vers l'appariement exhaustif relève le plafond de rappel sans coût supplémentaire, puisque le filtre est gratuit. Disposer de méthodes rapides et fiables de construction de graphes de connaissance peut aider au croisement de diverses sources pour le renseignement, et à la bonne utilisation de ces informations pour la prise de décision.

Références

[1] P. Degez, L. Jourdain, « Hybrid Canonicalization of Intelligence Knowledge Graphs: A Frugal and Explainable Approach », *CAID 2025*. — [2] B. Zhang, H. Soh, « Extract, Define, Canonicalize », *EMNLP 2024*. — [3] B. Mo et al., « KGGen: Extracting Knowledge Graphs from Plain Text with Language Models », *NeurIPS 2025*. — [4] A. Fader, S. Soderland, O. Etzioni, « Identifying Relations for Open Information Extraction », *EMNLP 2011*. — [5] U. Zaratiana et al., « GLiNER », *NAACL 2024*. — [6] F. Gilardi, M. Alizadeh, M. Kubli, « ChatGPT Outperforms Crowd Workers for Text-Annotation Tasks », *PNAS* 120(30), 2023. — [7] K. W. Kim, J.-D. Kim, A. Aizawa, « Refining and Reusing Annotation Guidelines for LLM Annotation », *ACL 2026*. — [8] S. Yao et al., « ReAct: Synergizing Reasoning and Acting in Language Models », *ICLR 2023*. — [9] M. Yuksekgonul et al., « TextGrad: Automatic "Differentiation" via Text », arXiv:2406.07496, 2024.