Généraliser l'adaptation de modèles de langue frugaux pour l'extraction de motifs RDF à partir de texte, à des relations de type Datatype et Object property

Célian Ringwald, Fabien Gandon, Catherine Faron, Franck Michel, Hanna Abi Akl

Généraliser l'adaptation de modèles de langue frugaux pour l'extraction de motifs RDF à partir de texte, à des relations de type *Datatype* et *Object property*

Célian Ringwald*, Fabien Gandon*
Catherine Faron*, Franck Michel* Hanna Abi Akl*

* Université Côte d'Azur, Inria, CNRS, I3S, Sophia-Antipolis, France
prenom.nom@inria.fr,
https://team.inria.fr/wimmics

Résumé. Les petits modèles de langage ont démontrés de bonne performances pour l'extraction de relations RDF à partir de shapes SHACL. Cet article, issu de notre travail accepté à K-CAP 2025, étudie leur capacité à traiter conjointement les propriétés de type Datatype et Object Property. Le principal défi réside dans l'extraction de propriétés rares. Pour y remédier, nous explorons plusieurs stratégies : échantillonnage stratifié, pondération de la perte, redimensionnement des données et génération synthétique par patrons. Les meilleurs résultats sont obtenus lorsque chaque propriété atteint un seuil minimal d'occurrences dans les données d'apprentissage. Nos données, résultats et code sont rendus publics afin d'assurer la reproductibilité. Ce travail propose ainsi des méthodes concrètes pour l'entraînement de SLM spécialisés et ouvre des perspectives pour l'extraction de relations sémantiques.

1 Introduction

Jusqu'à présent, la majorité des travaux relatif à la tâche d'extraction de relation (ER) se sont concentrés sur les relations fréquentes et sur des types d'entités bien définis et restreints. Des jeux de données de référence, tels que TACRED Zhang et al. (2017), T-REX Elsahar et al. (2018) et DocRED Yao et al. (2019), ont été proposés pour mener cette tâche à plus grande échelle, souvent fondés sur une supervision distante alignant le contenu textuel sur des graphes de connaissances existants. Ces approches ont permis de développer des modèles performants : Huguet Cabot et Navigli (2021); Josifoski et al. (2022), capables d'extraire de nombreuses relations impliquant des classes d'entités hétérogènes. D'autres travaux ont proposé de modéliser la tâche d'extraction de relations en suivant un schéma de données spécifique (Caufield et al. (2024), Mihindukulasooriya et al. (2023)). Néanmoins, l'extraction de relations rares ou complexes demeure un défi majeur pour les modèles de langage et reste très peu étudiée dans ce contexte.

Définition de la tâche : En suivant la formulation proposée dans Ringwald et al. (2025), nous apprenons des extracteurs de motifs RDF $\mathcal{M}_{Db}$ en ajustant de petits modèles de langage (Small Language Models, SLMs). L'entraînement repose sur une base duale $Db \subseteq W \times G$, où

W représente l'ensemble des textes et G l'ensemble des graphes de connaissances associés. La cible d'extraction est définie par une forme (shape) SHACL s. L'extracteur est une fonction :

$$E_{Db} : W \times S \to G; \quad (t, s) \mapsto \hat{g}$$

où $t \in W$ est un texte d'entrée, $s \in S$ un ensemble de formes SHACL, et $\hat{g}$ un graphe RDF dérivé de t et conforme à s. À partir de s, nous déduisons l'ensemble Π des motifs possibles (i.e. les combinaisons de propriétés définies par la forme). Chaque combinaison $\pi \in \Pi$ correspond à un *motif spécifique à un exemple*. Un graphe RDF g est valide par rapport à un motif π s'il contient exactement les propriétés de π et aucune autre.

Dans Ringwald et al. (2025), nous nous sommes concentrés sur l'extraction des propriétés de type *DataType*. Dans cet article, nous étendons cette approche à des formes SHACL intégrant également des *Object Properties*, qui correspondent à des relations reliant deux entités distinctes du graphe RDF, par opposition aux *Datatype Properties* qui associent une entité à une valeur littérale (par exemple une date ou un nombre). Nous soulignons en outre le défi que constitue le maintien d'une performance équilibrée sur l'ensemble des propriétés, y compris celles sous-représentées, et proposons plusieurs stratégies pour y remédier.

En résumé, cet article s'articule autour des deux questions de recherche suivantes :

— **RQ1.** Est-il possible de réaliser une extraction fondée sur des shapes SHACL contenant à la fois des *DataType Properties* et des *Object Properties* ?

— **RQ2.** Comment les extracteurs fondés sur les formes SHACL se comportent-ils lorsque la distribution des propriétés dans les jeux d'apprentissage est déséquilibrée ?

Nous étendons ainsi l'extraction guidée par des patrons RDF et des formes SHACL aux propriétés *datatype* et *object*. Pour gérer le déséquilibre et la longue traîne des propriétés, nous comparons plusieurs stratégies de rééquilibrage (pondération, stratification, augmentation). Nos contributions incluent cette extension, l'évaluation comparative des stratégies de rééquilibrage, et la mise à disposition de ressources ouvertes [1].

2 Extraction de Datatype et Object Properties

Distillation de la base de connaissances Dans Ringwald et al. (2025), nous avons procédé à la distillation d'une **base duale initiale** $\mathcal{K}$ issue du *datadump* DBpedia 2022.09 [2], contenant 6 109 994 résumés Wikipédia et leurs graphes DBpedia associés :

$$\begin{aligned} \mathcal{K} := \{(w, g) \in \mathcal{W} \times \mathcal{G}, \exists\, e \in IRI \text{ tel que} \\ desc_{\mathcal{W}}(e) = w \ \wedge \ desc_{\mathcal{G}}(e) = g\} \end{aligned} \tag{1}$$

Nous étendons ici la forme maximale décrivant `dbo:Person` dans Ringwald et al. (2025) [3], qui considérait 7 propriétés de type *DataType Property*. Nous ajoutons désormais trois relations de type *Object Properties* les plus courantes pour décrire les instances de `dbo:Person` : $\mathcal{P}(s^*_{op}) = \{\texttt{dbo:birthPlace}, \texttt{dbo:nationality}, \texttt{dbo:deathPlace}\}$
Ainsi, la nouvelle forme maximale combine les deux types de propriétés :

$$\mathcal{P}(s^*) = \mathcal{P}(s^*_{dt}) \cup \mathcal{P}(s^*_{op})$$

1. Kastor GitHub ; Zenodo repository
2. Jeu de données DBpedia utilisé
3. dbo:Person shape

Comme dans Ringwald et al. (2025), nous enrichissons les données afin de compenser les valeurs manquantes des propriétés de données à l'aide de règles adaptées *DataType Property* : notées $\mathcal{R}_{dt}$ [4], mais étendons aussi ces règles aux propriétés de type *Object*, notées $\mathcal{R}_{op}$ [5].

L'application des règles $\mathcal{R} = \mathcal{R}_{dt} \cup \mathcal{R}_{op}$ au graphe de connaissances initial produit une version enrichie :

$$\begin{aligned} \mathcal{K}^{\mathcal{R}} := \{(w, g')\,,\ (w, g) \in \mathcal{K} \\ \text{où } g' \text{ est le résultat de l'application de } \mathcal{R} \text{ à } g\} \end{aligned} \tag{2}$$

L'étape de post-traitement (*Wikicheck*) valide uniquement les triplets dont l'existence est vérifiée par des preuves textuelles issues du résumé Wikipédia correspondant. Initialement limitée aux *Data properties* via une recherche dans le texte brut (dans $\mathcal{W}_{plain}$), cette vérification a été étendue aux *Object properties* grâce à la conversion des pages Wikipédia en Markdown, permettant de contrôler la présence des URI et leur conformité avec les contraintes range définies dans la forme SHACL (dans $\mathcal{W}_{MD}$). En raison des restrictions d'accès à l'API, cette procédure a été appliquée à un sous-échantillon aléatoire d'entités dbo :Person pour éviter tout biais de sélection. Ce processus aboutit à une version distillée $\mathcal{K}_{dist}(s^*)$ contenant 136 718 graphes :

$$\begin{aligned} \mathcal{K}_{dist}(s^*) \coloneqq \{(w, w_{MD}, g); (w, g) \in \mathcal{K}^{\mathcal{R}}\,,\ w_{MD} \models g, \\ \exists \pi \in \Pi(s^*)\ ; g \text{ est valide vis-à-vis de } \pi\} \end{aligned} \tag{3}$$

où w_{MD} désigne la version Markdown de w.

Jeux de données La base $\mathcal{K}_{dist}(s^*)$ est utilisée pour échantillonner les jeux d'entraînement destinés à l'ajustement fin de modèles de langage préentraînés. Comme notre apprentissage s'appuie sur le modèle BART (publié en 2021), nous sélectionnons uniquement les articles Wikipédia publiés avant cette date. Nous imposons également que chaque graphe contienne au moins une relation de type *DataType property* et une relation de type *Object property*. Nous notons ce jeu de données $D1$:

$$\begin{aligned} D1 := \{(w, w_{MD}, g) \in \mathcal{K}_{dist}(s^*); \\ g \text{ contient au moins une propriété de } s^*_{op} \\ \text{et au moins une propriété de } s^*_{dt} \\ \text{et } w \text{ a été créé avant 2021}\} \end{aligned} \tag{4}$$

Modèle de référence : REBEL Pour comparer notre méthode à l'état de l'art, nous confrontons nos résultats à ceux de *REBEL-large* (406 M de paramètres), un modèle de référence pour l'extraction de relations. Comme REBEL n'est pas conçu pour l'extraction de triplets RDF, nous avons adapté ses sorties pour une comparaison équitable. Nous avons aligné les propriétés extraite par REBEL sur celles de DBpedia pour définir la forme s^*_{rebel}, et ajouté les triplets inférés par $\mathcal{R}_{dt}$ et $\mathcal{R}_{op}$. Enfin REBEL ne générant pas d'URI, nous avons utilisé dans un second temps DBpediaLookup [6] pour lier les labels produits par REBEL à des URIs referés dans DBpedia.

4. Voir les détails sur les règles associées aux *DataType Property*
5. Voir les détails sur les règles associées aux *Object Properties*
6. `https://github.com/dbpedia/dbpedia-lookup`

TAB. 1 – *Fréquence et nombre de triplets par propriété dans la base distillée et dans l'échantillon $D1$*

	$\mathcal{K}_{dist}(s^*)$		$D1$	
prop	Nb	Fréq.	Nb	Fréq.
birthYear	105,818	0.77	1,010	0.84
birthPlace	15,279	0.11	996	0.83
birthDate	97,039	0.71	927	0.77
label	92,691	0.68	870	0.73
deathYear	37,633	0.28	457	0.38
deathDate	32,743	0.24	395	0.33
deathPlace	4,859	0.04	291	0.24
nationality	2,211	0.02	162	0.14
birthName	12,265	0.09	130	0.11
alias	1,398	0.01	14	0.01

Détails d'apprentissage Tous les modèles sont issus du finetuning de BART-base (140M de paramètres). Les graphes sont linéarisés selon la syntaxe « TurtleLight one line and factorised » Ringwald et al. (2025). L'entraînement est réalisé sur un GPU Nvidia Tesla V100 avec un taux d'apprentissage initial de 0,00005, 1 000 étapes de "warmup" et un arrêt anticipé après 5 itérations sans amélioration. Chaque modèle est finetuné à l'aide d'une validation croisée à 10 plis, le details des modèles obtenus est disponible sur WanDB [7].

Évaluation et métriques Nous utilisons la comparaison stricte entre valeurs attendues et générées pour calculer le macro-F1 (F_1^+) et le micro-F1 (F_1^-). Le micro-F1 reflète la performance globale (pondérée par la fréquence des propriétés), tandis que le macro-F1 donne une vision équilibrée tenant compte des propriétés rares. Nous rapportons la moyenne et l'écart-type des mesures sur les 10 plis.

TAB. 2 – *Performances des modèles $\mathcal{M}D1$ adaptés à différentes shapes and type d'entrées*

Model	$Recall$	$Prec.$	F_1^-	F_1^+
$\mathcal{M}D1(MD, s^*_{dt})$	98 ± 2	89 ± 3	97 ± 3	89 ± 11
$\mathcal{M}D1(MD, s^*_{op})$	91 ± 13	88 ± 13	90 ± 13	83 ± 22
$\mathcal{M}D1(MD, s^*)$	95 ± 5	92 ± 6	94 ± 5	86 ± 14
$\mathcal{M}D1(PLAIN, s^*_{dt})$	98 ± 2	96 ± 3	98 ± 3	91 ± 10
$\mathcal{M}D1(PLAIN, s^*_{op})$	90 ± 14	87 ± 15	89 ± 14	83 ± 22
$\mathcal{M}D1(PLAIN, s^*)$	96 ± 4	94 ± 6	95 ± 5	88 ± 14

Résultats et analyse Les performances sont résumées dans la Table 2. On observe que les modèles basés sur du texte Markdown n'apportent pas d'amélioration notable pour les propriétés d'objet, probablement à cause du bruit généré par les URI. Les scores F1 sont plus faibles pour sur les *Object Properties* (s^*_{op}) que sur les *DataType Properties* (s^*_{dt}), avec une variance plus élevée, signe d'une instabilité accrue. Pour tester l'impact de la fréquence des propriétés, nous distinguons deux sous-ensembles : s^*_+ (propriétés fréquentes) et s^*_- (propriétés rares). Les résultats (Table 3) montrent que la fréquence influence fortement la performance et que spécialiser un modèle sur les propriétés rares ne suffit pas à compenser leur rareté.

REBEL ne produit que 70 % des URI attendues pour se référencer aux sujets des relations lorsqu'il est combiné à un post-traitement de liage d'entités (*entity linking*). En revanche, nos modèles génèrent des triplets RDF valides à 100 %, se référant systématiquement aux URI des sujets attendus. De plus, comme le montre la Table 4, nos modèles surpassent REBEL tant en précision structurelle qu'en capacité de généralisation. À ce propos, il convient de

7. https://wandb.ai/celian-ringwald/GenLimits_Part1

TAB. 3 – *Performances des modèles $\mathcal{MD}1$ adapté aux nouveaux sets de propriétés proposés*

$Model$	F_1^-	F_1^+
$\mathcal{MD}1(PLAIN, s^*)$	95 ± 5	88 ± 15
$\mathcal{MD}1(PLAIN, s^*_+)$	$\mathbf{96 \pm 4}$	$\mathbf{96 \pm 5}$
$\mathcal{MD}1(PLAIN, s^*_-)$	$\underline{70} \pm 21$	$\underline{64 \pm 26}$

souligner que finetuner un modèle en suivant notre méthodologie ne requière que 10 minutes d'entraînement (cf. Ringwald et al. (2025)), alors le finetuning de REBEL en requière plus de 27 heures.

TAB. 4 – *Comparaison de $\mathcal{MD}1$ avec REBEL*

Model	$Recall$	$Prec.$	F_1^-	F_1^+
$\mathcal{MD}1(PLAIN, s^*_{rebel})$	96 ± 5	93 ± 5	94 ± 5	89 ± 10
$REBEL$	54 ± 1	83 ± 1	65 ± 1	41 ± 1
$REBEL \times DBpediaLookup$	59 ± 0.3	83 ± 1	69 ± 0.3	44 ± 0.6

3 Défis liés à la distribution en longue traîne des propriétés

Les expériences menées sur un ensemble élargi de 16 formes issues des micro-ontologies définies par le Text2KGBench [8] ont aussi démontrés de fortes variations en terme de F_1, attribuées au déséquilibre entre propriétés fréquentes et rares. Nous explorons ici plusieurs solutions visant à atténuer cet effet sur la forme `dbo:Person`. Plus de details concernant les modèles obtenus sont disponibles sur WanDB [9].

3.1 Apprentissage à grande échelle et rééquilibrage des propriétés rares

Nous étudions ici trois leviers visant à améliorer la robustesse des modèles face au déséquilibre des propriétés.
(1) Apprentissage à grande échelle : Jusqu'à présent, l'entraînement reposait sur de petits jeux de données offrant un bon compromis entre performance et coût. Nous évaluons désormais l'effet du passage à l'échelle à l'aide de trois jeux plus volumineux — $D2$, $D4$ et $D10$ — contenant respectivement 2 400, 4 800 et 12 000 exemples. Un quatrième jeu, D_-, cible spécifiquement les propriétés rares (s^*_-).
(2) Stratification ciblée : Le déséquilibre entre propriétés peut fausser les performances selon les plis de validation. Pour y remédier, nous introduisons une stratification focalisée sur les propriétés rares, suivant trois règles : (a) un graphe contenant une seule propriété rare est affecté à la strate correspondante ; (b) s'il en contient plusieurs, il est affecté à la moins représentée ; (c) en l'absence de propriété rare, il rejoint la strate *Other*. Les modèles sont entraînés sur cinq plis afin d'augmenter les chances qu'un split contienne au moins quelques examples provenant d'une classe rare. Pour plus de details [10].
(3) Perte pondérée par propriété : Pour renforcer l'apprentissage des propriétés sous-représentées, nous pondérons la fonction de perte. La Cross-Entropy standard est modifiée par un poids ω_y dépendant de la fréquence inverse de la strate associée à chaque séquence. Cette pondération

8. `https://wandb.ai/celian-ringwald/Text2KGBench_KastorModels`
9. `https://wandb.ai/celian-ringwald/GenLimitsPart2_train`
10. `https://github.com/datalogism/Kastor/blob/main/doc/SamplingStrategies.md`

confère davantage d'importance aux propriétés rares tout en restant compatible avec le cadre séquence-à-séquence.
(4) Évaluation croisée : Pour évaluer la généralisation, nous utilisons quatre jeux indépendants (200 exemples chacun) : d_N — articles Wikipedia créés après la puiblication de BART ; d_+ — exemples comportant uniquement des propriétés fréquentes ; d_- — exemples contenant au moins une propriété rare ; d_R — échantillon aléatoire mixte. Ces jeux garantissent une comparaison équitable entre les modèles et permettent d'évaluer leur capacité à généraliser à de nouveaux contextes.

TAB. 5 – *Scores F1 obtenus par chaque modèle sur les quatre jeux de données de validation croisée. Les métriques sont moyennées sur l'ensemble des plis, et l'écart-type est toujours inférieur à 5 points. Les meilleurs scores sont indiqués en gras, et les moins bons sont soulignés.*

Model	Config	F_1^-				F_1^+			
		d_N	d_+	d_-	d_R	d_N	d_+	d_-	d_R
$\mathcal{M}D1(PLAIN, s^*)$	CE (10-folds)	82.31	91.71	76.90	83.85	63.07	96.72	59.19	61.98
$\mathcal{M}D\_(PLAIN, s^*)$	CE (10-folds)	72.66	76.77	**95.09**	75.49	57.48	84.86	**90.68**	58.13
$\mathcal{M}D2(PLAIN, s^*)$	CE (10-folds)	82.08	92.00	77.57	84.58	62.90	96.65	61.10	62.66
$\mathcal{M}D4(PLAIN, s^*)$	CE (10-folds)	84.19	91.88	79.20	86.39	64.18	**97.06**	63.34	65.68
$\mathcal{M}D10(PLAIN, s^*)$	CE (10-folds)	**84.42**	**91.94**	86.05	**86.81**	**67.05**	97.05	**77.26**	66.52
$\mathcal{M}D10_{Strat}(PLAIN, s^*)$	CE-STRAT (5-folds)	83.87	90.06	83.56	86.45	65.77	95.29	71.92	**67.83**
$\mathcal{M}D10_{StratWCE}(PLAIN, s^*)$	WCE-STRAT (5-folds)	84.16	91.26	84.53	86.49	66.02	96.20	74.93	66.51

Résultats et analyse Les résultats (Table 5) montrent que ni la stratification ni la perte pondérée n'améliorent significativement les performances. En revanche, l'augmentation de la taille du jeu d'entraînement accroît nettement la robustesse des modèles, avec un gain d'environ 10 points de F_1 sur les propriétés rares (d_-) et une amélioration globale sur toutes les configurations. Les modèles conservent des performances élevées sur les propriétés fréquentes (d_+) et restent stables sur les textes inédits (d_N). L'analyse fine de nos résultats montre que le passage à grande échelle ($\mathcal{M}D10$) favorise la couverture des propriétés rares, mais qu'une spécialisation extrême ($\mathcal{M}D\_$) demeure plus performante sur ces dernières. Aucun modèle n'extrait correctement la propriété rare `dbo:alias`, illustrant la persistance de la longue traîne malgré la montée en échelle sur l'échantillon aléatoire d_R.

3.2 Atteindre une exposition suffisante par propriété

Stratégies d'augmentation de données Afin d'améliorer la couverture des propriétés rares, notamment `dbo:alias`, nous explorons trois stratégies d'augmentation.
(1) L'utilisation exhaustive des entités disponibles enrichit le jeu $D10$ en incluant tous les exemples contenant cette propriété ($D10A_{all}$).
(2) La création d'examples synthétique générés à partir de gabarits d'abstracts afin d'atteindre le seuil de 1 000 occurrences pour chaque propriété visée : ($D10A_{KR0}$ et $D10A_{KR1}$). Pour plus de details [11].
(3) Enfin, la construction d'un jeu équilibré ($D4SE_{all}$) garantissant un minimum de 1 000 exemples par propriété via un échantillonnage aléatoire contrôlé.
Ces méthodes permettent d'assurer une exposition suffisante pour chaque propriété tout en limitant la taille globale du corpus. Les details de l'entrainement et de la validation de ces modèles sont disponibles sur Wandb [12].

11. `https://github.com/datalogism/Kastor/blob/main/doc/AugStrategies.md`
12. Entrainement : `https://wandb.ai/celian-ringwald/GenLimitsPart3_train`, Evaluation croisée :`https://wandb.ai/celian-ringwald/GenLimitsPart3_crossEval`

TAB. 6 – *Scores F1 obtenus par chaque modèle sur les cinq jeux de données de validation croisée. Les métriques sont moyennées sur l'ensemble des plis, et l'écart-type est toujours inférieur à 5 points. Les meilleurs scores sont indiqués en gras, et les moins bons sont soulignés.*

Model	F_1^-					F_1^+				
	d_N	d_+	d_-	d_R	d_C	d_N	d_+	d_-	d_R	d_C
$\mathcal{MD}10(PLAIN, s^*)$	84.42	**91.94**	**86.05**	86.81	72.46	67.05	97.05	77.26	66.52	60.97
$\mathcal{MD}10A_{all}(PLAIN, s^*)$	84.00	91.85	85.15	**87.19**	72.36	67.44	96.38	76.65	69.54	63.92
$\mathcal{MD}10A_{KR0}(PLAIN, s^*)$	84.00	91.70	82.07	86.71	71.29	64.57	95.72	71.22	65.49	58.26
$\mathcal{MD}10A_{KR1}(PLAIN, s^*)$	**86.85**	91.85	85.02	86.85	72.47	**67.30**	**97.11**	71.22	**85.02**	59.64
$\mathcal{MD}4SE_{all}(PLAIN, s^*)$	76.41	79.05	85.69	79.66	**79.90**	65.07	89.23	**89.24**	64.62	**72.74**

Résultats L'évaluation repose sur les jeux introduits à la section 3.1, auxquels s'ajoute un jeu corrigé d_C issue de l'annotation manuelle de 1 470 erreurs (FP/FN) et permettant de mesurer sans biais les modèles. Les résultats (Table 6) montrent que le modèle $\mathcal{MD}4SE_{all}$ surpasse les autres configurations avec un gain d'environ 10 points de F1, confirmant l'efficacité de l'exposition équilibrée. Les analyses par propriété indiquent que $\mathcal{MD}4SE_{all}$ et $\mathcal{MD}10A_{all}$ améliorent la détection des propriétés rares (`dbo:deathPlace`, `dbo:birthName`, `dbo:alias`). Sur le jeu aléatoire d_R, seules ces deux configurations parviennent à extraire correctement `dbo:alias`. Enfin, l'évaluation sur d_C montre que plusieurs erreurs initialement classées comme fausses positives relèvent en réalité de découvertes de faits valides, faisant de $\mathcal{MD}4SE_{all}$ le modèle le plus robuste face aux propriétés peu représentées.

4 Discussion et conclusion

Nos expériences confirment la capacité des modèles proposés à extraire efficacement des motifs RDF à partir d'une seule forme SHACL intégrant des DataType et Object Properties, avec de bonnes performances sur les propriétés fréquentes. Des SLMs spécialisés, plus légers, surpassent même des modèles génériques comme REBEL tout en restant économes en ressources. L'étude montre aussi que le format Markdown et les pondérations de la perte n'apportent pas d'amélioration notable, tandis qu'une exposition minimale de 1 000 occurrences par propriété est essentielle pour une performance stable. Ces résultats soulignent l'importance d'un échantillonnage équilibré et ouvrent la voie à des travaux futurs sur de nouvelles métriques pour les Object Properties et sur des stratégies avancées d'augmentation de données.

Remerciements. Cette recherche a été soutenue par 3IA Côte d'Azur Investments (ANR-23-IACL-0001), UCAJEDI (ANR-15-IDEX-01), l'infrastructure OPAL et le Centre de calcul haute performance de l'Université Côte d'Azur. La relecture grammaticale a été réalisée avec l'aide de ChatGPT (GPT-4) et Grammarly. Tous les contenus générés par l'IA ont été vérifiés, corrigés et validés par les auteurs, qui assument l'entière responsabilité du manuscrit final.

Références

Caufield, J. H., H. Hegde, V. Emonet, N. L. Harris, M. P. Joachimiak, N. Matentzoglu, H. Kim, S. Moxon, J. T. Reese, M. A. Haendel, P. N. Robinson, et C. J. Mungall (2024). Structured prompt interrogation and recursive extraction of semantics (spires) : a method for populating knowledge bases using zero-shot learning. *Bioinformatics 40*(3).

Elsahar, H., P. Vougiouklis, A. Remaci, C. Gravier, J. Hare, F. Laforest, et E. Simperl (2018). T-REx : A large scale alignment of natural language with knowledge base triples. In *Pro-*

ceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), Miyazaki, Japan. European Language Resources Association (ELRA).

Huguet Cabot, P.-L. et R. Navigli (2021). REBEL : Relation extraction by end-to-end language generation. In *Findings of the Association for Computational Linguistics : EMNLP 2021*, Punta Cana, Dominican Republic, pp. 2370–2381. ACL.

Josifoski, M., N. De Cao, M. Peyrard, F. Petroni, et R. West (2022). GenIE : Generative information extraction. In *Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, Seattle, United States, pp. 4626–4643. ACL.

Mihindukulasooriya, N., S. Tiwari, C. F. Enguix, et K. Lata (2023). Text2kgbench : A benchmark for ontology-driven knowledge graph generation from text. In *Proceedings ISWC 2023*, Cham, pp. 247–265. Springer.

Ringwald, C., F. Gandon, C. Faron, F. Michel, et H. A. Akl (2025). Kastor : Fine-tuned small language models for shape-based active relation extraction. In *European Semantic Web Conference*, Volume 15718 of *Lecture Notes in Computer Science*, pp. 94–115. Springer, doi: 10.1007/978-3-031-94575-5_6.

Yao, Y., D. Ye, P. Li, X. Han, Y. Lin, Z. Liu, Z. Liu, L. Huang, J. Zhou, et M. Sun (2019). DocRED : A large-scale document-level relation extraction dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy, pp. 764–777. Association for Computational Linguistics, doi: 10.18653/v1/P19-1074.

Zhang, Y., V. Zhong, D. Chen, G. Angeli, et C. D. Manning (2017). Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*.

Summary

Small Language Models (SLMs) demonstrate strong effectiveness in RDF relation extraction guided by SHACL shapes. This paper, based on our work accepted at K-CAP 2025, investigates their ability to jointly handle both Datatype and Object Property relations. The main challenge lies in extracting rare properties. To address this issue, we explore several strategies, including stratified sampling, loss weighting, data scaling, and pattern-based synthetic data generation. The best results are achieved when each property reaches a minimal occurrence threshold within the training data. To ensure reproducibility, we publicly release all datasets, experimental results, and source code. This work thus provides practical methods for training specialized SLMs and highlights promising directions for semantic relation extraction.