KGPRUNE : une application Web pour extraire des sous-graphes d'intérêt de Wikidata par élagage analogique

Pierre Monnin[1], Cherif-Hassan Nousradine[2], Lucas Jarnac[2,3], Laurel Zuckerman[4], Miguel Couceiro[2,5]

Les graphes de connaissance (GC, en anglais *knowledge graphs* ou KG) sont des représentations structurées qui modélisent les connaissances d'un ou plusieurs domaines. Leurs unités atomiques sont des triplets (h, p, o) qui représentent l'existence d'une relation p entre deux entités h et o.

De par leur flexibilité et les connaissances qu'ils fournissent, les GC sont devenus des atouts majeurs et alimentent diverses méthodes d'intelligence artificielle (par exemple, *retrieval augmented generation* pour les grands modèles de langages [19], machine learning en général [4]) avec des applications dans un large éventail de domaines (recherche, e-commerce, réseaux sociaux, sciences de la vie, etc. [3, 9, 18]).

Suivant un des principes fondateurs du Web sémantique [1], les GC existants sont souvent réutilisés pour créer de nouveaux GC ou pour soutenir de nouvelles tâches ou applications [5, 7, 15, 22]. Cette réutilisation est rendue possible par

1. Université Côte d'Azur, Inria, CNRS, I3S, Sophia-Antipolis, France, pierre.monnin@inria.fr.
2. Université de Lorraine, CNRS, LORIA, Nancy, France.
3. Orange, France.
4. Chercheur indépendant.
5. Universidade de Lisboa, INESC-ID, Instituto Superior Técnico, Lisboa, Portugal.

DOI:10.48556/SIF.1024.24.177

le nombre et la taille croissants des GC publiquement accessibles[6], et en particulier des grands GC couvrant plusieurs domaines tels que Wikidata. Ce dernier est un GC générique de plus de 100 millions de nœuds[7] qui soutient Wikipédia [24]. Wikidata est considéré comme une source de connaissances de premier ordre, mais plusieurs problèmes entravent sa réutilisation [11, 21]. Premièrement, sa grande taille entraîne des problèmes de scalabilité lors de la manipulation du graphe (stockage, performances des requêtes, etc.). Deuxièmement, toutes les connaissances représentées ne sont pas pertinentes pour chaque tâche ou application considérée. Par exemple, l'une des entités voisines de `Microsoft SharePoint` est `Dating App` qui peut ne pas être intéressante lors de la création d'un GC d'entreprise modélisant le domaine de l'informatique d'entreprise.

Pour résoudre ces problèmes, plusieurs auteurs ont proposé d'extraire des sous-graphes, soit manuellement avec les premiers exemples remontant à 1996 [22] soit automatiquement [10, 11, 21]. En particulier, nous avons récemment proposé une approche qui parcourt le voisinage d'entités de départ fournies par les utilisateurs, en conservant les voisins pertinents tout en élaguant ceux qui ne le sont pas [10]. Cette approche s'appuie sur l'inférence analogique et présente des performances élevées, y compris dans des situations de transfert, avec un nombre de paramètres particulièrement faible.

En nous appuyant sur ce travail précédent, nous proposons KGPrune,[8] une application Web permettant d'extraire des sous-graphes de Wikidata à partir d'entités de départ et de propriétés intéressantes pour l'utilisateur. KGPrune peut être utilisée à la fois depuis un navigateur et via une API, permettant aux utilisateurs avec diverses expertises techniques d'interagir avec notre approche d'élagage. Dans ce qui suit, après avoir décrit les fonctionnalités et l'architecture technique de KGPrune, nous illustrons son intérêt avec deux applications concrètes : l'amorçage d'un graphe de connaissances d'entreprise et l'extraction de connaissances liées aux œuvres d'art pillées. Une vidéo de démonstration est disponible sur YouTube.[9]

KGPrune : extraire des sous-graphes d'intérêt

Les principaux écrans de l'application Web KGPrune sont présentés en figure 1. Nous décrivons ci-dessous les principales caractéristiques et étapes d'interaction avec l'application.

6. `https://lod-cloud.net/`.
7. `https://www.wikidata.org/wiki/Wikidata:Statistics`.
8. `https://kgprune.loria.fr`.
9. `https://youtu.be/mt5gF4ZmhGY`.

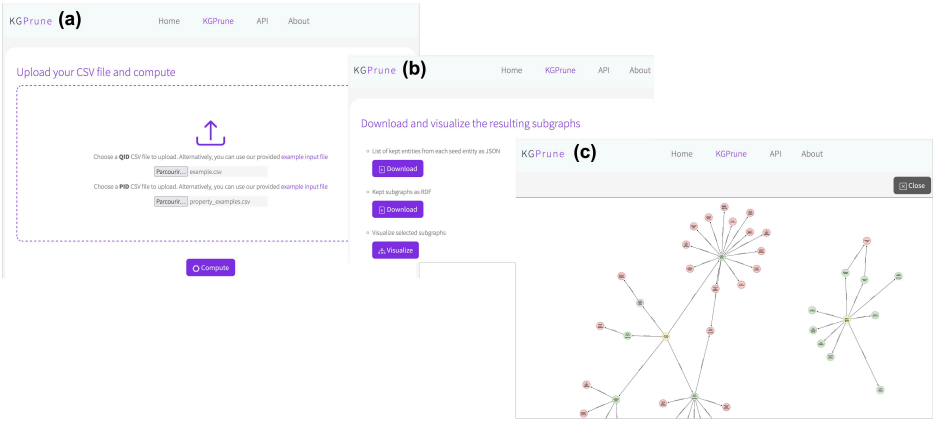

Fig. 1. Écrans principaux de KGPrune. (a) Formulaire de soumission des deux fichiers CSV demandés à l'utilisateur : l'un indiquant les QIDs des entités de départ et l'autre indiquant les PIDs des propriétés à parcourir. (b) Page d'affichage des résultats où l'utilisateur peut choisir de visualiser les sous-graphes extraits ou de les télécharger en JSON ou RDF. (c) Visualisation des sous-graphes extraits où les entités de départ sont en jaune, les voisins conservés en vert et les voisins élagués en rouge.

GC supporté.

Nous avons choisi de construire KGPrune sur Wikidata étant donné sa taille et sa généricité. Il peut donc servir de source de connaissances de premier ordre pour plusieurs domaines. Il convient toutefois de noter que notre approche pourrait être appliquée à n'importe quel GC.

Fichiers en entrée.

KGPrune ne nécessite que deux fichiers CSV comme entrées de l'utilisateur, comme illustré dans la table 1. Le fichier `qid_example.csv` contient des QIDs identifiant les entités de départ d'intérêt dont le voisinage sera récupéré. Ici, à titre d'exemple, nous considérons `Microsoft SharePoint` (Q18833) et le langage de programmation `Java` (Q251). Le fichier `pid_example.csv` contient des PIDs identifiant les propriétés d'intérêt dont les arêtes seront traversées. Ici, nous considérons `instance of` (P31), `subclass of` (P279) et `part of` (P361). L'indication du PID d'une propriété conduit à traverser des arêtes directes tandis que l'indication (-)PID conduit à traverser des arêtes inverses. Ici, les arêtes directes et inverses de P279 seront parcourues. L'écran d'entrée de KGPrune est présenté sur la figure 1.a.

qid_example.csv	pid_example.csv
Q18833	P31
Q251	P279
	(-)P279
	P361

Table 1. Exemple de fichiers d'entrée pour KGPrune. Le fichier `qid_example.csv` contient les QIDs des entités de départ d'intérêt. Leur voisinage sera récupéré en parcourant les arêtes étiquetées par les propriétés dont les PIDs sont spécifiés dans `pid_example.csv`.

Extraction de sous-graphes.

Une fois les fichiers CSV d'entrée soumis, KGPrune exécute notre algorithme de parcours et d'élagage [10][10]. À partir des entités de départ, les arêtes étiquetées par les propriétés spécifiées sont parcourues. Pour chaque voisin, notre modèle d'élagage analogique décide de le conserver ou de l'élaguer. Les analogies sont des énoncés de la forme A est à B ce que C est à D , modélisés comme des quadruplets $A : B :: C : D$ tels que $\text{Paris} : \text{France} :: \text{Berlin} : \text{Allemagne}$. De tels quadruplets capturent les similarités et les dissimilarités entre les objets [16, 17]. Ici, étant donné une entité de départ e_s^u spécifiée par l'utilisateur et un de ses voisins e_r^u, notre modèle prédit s'ils forment une analogie avec une entité de départ e_s^k et un de ses voisins e_r^k pour lesquels une décision de conservation est connue :

$$\underbrace{e_s^k \ : \ e_r^k}_{\text{Décision de « conservation » connue}} \quad :: \quad \underbrace{e_s^u \ : \ e_r^u}_{\text{Décision inconnue}} \tag{1}$$

Cette prédiction repose sur des plongements de graphe préappris pour les entités et sur le modèle convolutionnel de détection d'analogie introduit par [13]. Grâce à son architecture, ce modèle est capable de capturer les similarités et les dissimilarités relatives entre les entités de départ et leurs voisins à conserver ou à élaguer, et est ainsi capable de généraliser à des entités hétérogènes non-vues à l'entraînement. Si notre modèle prédit que les quatre entités mentionnées précédemment forment une analogie, la décision connue entre e_s^k et e_r^k (i.e. garder e_r^k) est extrapolée à e_r^u. Sinon, e_r^u est élaguée. Essentielles au modèle, les décisions connues proviennent d'un ensemble de données annoté manuellement nommé `dataset1` et publiquement accessible.[11]

10. https://github.com/Orange-OpenSource/analogical-pruning.
11. https://doi.org/10.5281/zenodo.8091584.

Ce processus est effectué de manière itérative sur le voisinage des nœuds conservés jusqu'à ce qu'aucun autre voisin ne puisse être atteint. Les résultats sont ensuite affichés à l'utilisateur (figure 1.b) qui peut choisir de visualiser les sous-graphes extraits (figure 1.c) ou de les télécharger au format JSON ou RDF pour les importer dans un nouveau GC. L'interface de visualisation permet aux utilisateurs d'explorer les voisinages des entités de départ et d'évaluer les résultats de l'élagage. En particulier, les utilisateurs peuvent remarquer dans l'interface utilisateur si notre modèle a élagué à tort un voisin qui les intéresse et l'ajouter aux entités de départ pour forcer sa prise en compte. Cela ouvre la voie à un processus d'élagage itératif dans lequel les utilisateurs explorent les résultats de l'élagage et ajoutent de nouvelles entités de départ qui seront exploitées dans les itérations suivantes.

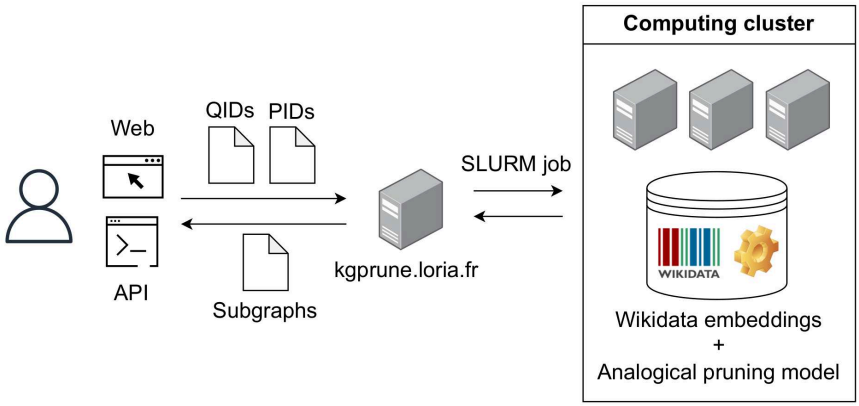

Fig. 2. Architecture technique de KGPrune. Les utilisateurs peuvent soumettre des tâches d'extraction de sous-graphes via le site Web ou l'API. Ces tâches sont ensuite soumises sous forme de tâches SLURM à notre cluster de calcul.

Architecture technique.

KGPrune repose sur l'architecture technique présentée sur la figure 2. Les utilisateurs peuvent interagir avec l'application via un navigateur Web ou une API. Leurs tâches d'extraction de sous-graphes sont envoyées sous forme de tâches SLURM à notre cluster de calcul où l'adjacence du graphe Wikidata, les plongements pré-entraînés, et nos modèles d'élagage analogique sont chargés et utilisés en inférence.

Pour l'apprentissage des plongements de Wikidata, nous avons utilisé le modèle TransE [2] avec une dimension de 200. Pour le modèle analogique, nous l'avons entraîné en utilisant `dataset1`, un des deux jeux de données que nous

avons préalablement annotés manuellement et rendus publiquement accessibles. Nous utilisons 16 filtres sur la première couche convolutionnelle et 8 filtres sur la deuxième couche convolutionnelle.

Notre modèle atteint des performances compétitives par rapport aux principaux modèles de l'état de l'art, avec un nombre de paramètres considérablement inférieur et une capacité de généralisation supérieure dans un contexte de transfert (table 2).

Modèle		**LSTM**	**Path Analogy**
dataset1	Précision	79.72 ± 5.17	80.10 ± 0.84
	Rappel	76.00 ± 6.59	74.44 ± 5.28
	F1	77.43 ± 2.38	77.06 ± 2.89
	ACC	83.48 ± 3.05	83.51 ± 2.87
	# paramètres	210,751	1,401
dataset2	Précision	78.49 ± 8.80	81.63 ± 8.27
	Rappel	94.58 ± 2.96	94.90 ± 2.16
	F1	85.36 ± 4.53	87.54 ± 5.05
	ACC	78.66 ± 5.95	82.50 ± 6.07
	# paramètres	210,751	251
Transfert	Précision	92.83	91.49
	Rappel	74.73	83.39
	F1	82.80	87.25
	ACC	80.04	84.33

Table 2. Performances de notre modèle comparé à un LSTM, son principal concurrent. Le contexte de transfert correspond à entraîner le modèle sur dataset1 et à le tester sur dataset2. Les expériences sur chaque jeu de données ont été réalisées en utilisant une validation croisée en 5 blocs et avec un hypertuning du nombre de filtres. Les résultats complets sont disponibles dans [10].

Deux cas d'usage illustratifs

Pour illustrer l'impact de l'application KGPrune, nous l'avons expérimentée sur deux cas d'utilisation : l'amorçage d'un GC d'entreprise et l'extraction de sous-graphes liés à des œuvres d'art pillées, attestant l'utilité de notre outil sur de réelles applications.

Amorçage d'un GC d'entreprise (GCE).

La construction d'un nouveau GC nécessite son amorçage avec un noyau de qualité qui peut ensuite soutenir des approches d'extraction automatique de connaissances à partir de données structurées ou non structurées (par exemple, tables, textes) [14, 20, 25]. Ces approches enrichissent ensuite le GC tout en étant guidées par les termes et les relations que le GC fournit, formant une boucle vertueuse.

Pour construire un tel noyau, plusieurs auteurs s'appuient sur Wikidata. Pour limiter la taille du noyau créé, des parties du voisinage des entités de départ d'intérêt sont sélectionnées avec un processus de distillation [21] ou d'élagage [10, 11]. Leurs parcours du graphe se concentrent sur la hiérarchie de l'ontologie, uniquement vers la racine [21] ou à la fois vers la racine et vers les feuilles [11].

Dans [10], nous avons proposé notre approche d'élagage pour amorcer un GCE centré sur le domaine de l'informatique, en parcourant l'ontologie vers la racine et vers les feuilles, à partir des entités de départ d'intérêt disponibles dans le glossaire interne de l'entreprise. Les performances obtenues par notre approche (table 2) sont très compétitives par rapport aux autres modèles d'état de l'art, avec un nombre de paramètres 100 à 1000 fois moindre. Cela illustre l'intérêt de notre approche pour ce cas d'utilisation. Avec KGPrune, nous avons étendu notre approche précédente en permettant à l'utilisateur de définir les propriétés à parcourir, enrichissant ainsi les sous-graphes extraits de Wikidata, et en offrant des capacités de visualisation pour permettre à l'utilisateur d'explorer les voisinages extraits.

Extraction de sous-graphes liés aux œuvres d'art pillées.

Les réseaux de pillage d'œuvres d'art opèrent à de nombreux niveaux et sur de longues périodes de temps. Certaines agences soulignent qu'il s'agit d'une industrie criminelle qui rapporte des milliards de dollars par an. Une documentation fiable est donc de la plus haute importance pour retrouver des biens culturels perdus ou volés et pour établir la propriété légitime. Il s'agit cependant d'une tâche difficile car les données sur le patrimoine culturel sont enfermées dans des silos de données, ce qui rend exceptionnellement difficiles la recherche, la localisation et l'obtention d'une documentation fiable [8].

Les auteurs de [8] proposent l'utilisation des données ouvertes et liées (en anglais, *Linked Open Data* ou LOD) comme base de données mondiale sur le patrimoine culturel. Ils ont exploré le potentiel des LOD pour intégrer de grandes quantités de données sur le patrimoine culturel, facilitant l'accès à l'information dans ce domaine, afin d'aider à protéger les biens culturels contre le pillage et de suivre les œuvres d'art pillées [26, 27]. Cependant, le suivi des œuvres d'art volées implique des connaissances relatives aux œuvres d'art, à la

généalogie, à la propriété, à la provenance. Certaines de ces connaissances sont présentes dans des GC génériques, mais associées à des connaissances non pertinentes pour la tâche considérée (e.g. biologie, informatique). Par exemple, Wikidata contient 43 730 marchands d'art, collectionneurs, conservateurs et galeries ; 6 648 musées d'art ; 930 405 peintures ; 251 personnes sur lesquelles l'Art Looting Investigation Unit[12] a enquêté ; et des propriétés telles que `propriétaire de` et `appartenant à`. D'où la nécessité d'extraire des sous-graphes spécifiques traitant de thèmes pertinents tout en évitant les informations fausses, inexactes ou non pertinentes.

Dans cette optique, KGPrune a le potentiel d'extraire et de collecter des informations fiables et pertinentes à partir de Wikidata. Dans des expériences préliminaires, nous avons appliqué notre approche sur le voisinage d'œuvres d'art connues (e.g. `Les Cyprès`), d'artistes (e.g. `Alexej von Jawlensky`), de musées (e.g., `National Gallery of Arts`) et de marchands d'art (e.g. `Alfred Flechtheim`). Les résultats ont montré une bonne adéquation avec les besoins humains en informations nécessaires au suivi des œuvres d'art volées. Nous sommes en train d'explorer plus en détail comment KGPrune, et en particulier ses fonctionnalités d'élagage et de visualisation, peuvent prendre en charge d'autres cas d'utilisation liés au patrimoine culturel.

Conclusion et perspectives

Dans cet article, nous avons présenté KGPrune, une application Web permettant aux utilisateurs d'extraire des sous-graphes d'intérêt de Wikidata en fournissant des entités de départ d'intérêt et des propriétés à parcourir. Notre application évite une potentielle dérive thématique lors du parcours du graphe en s'appuyant sur un mécanisme d'élagage efficace basé sur le raisonnement par analogie. Les utilisateurs peuvent interagir avec KGPrune via leur navigateur Web ou une API, facilitant ainsi son intégration dans des projets divers. Nous avons démontré l'intérêt de l'application avec deux cas d'utilisation concrets.

Actuellement, KGPrune ne prend en charge que Wikidata. Nous envisageons à l'avenir d'intégrer des GC supplémentaires (par exemple, DBpedia [12], YAGO [23], Bio2RDF [6]), fournissant de ce fait aux utilisateurs des contextes supplémentaires à partir desquels extraire des sous-graphes. De plus, notre modèle basé sur l'analogie est entraîné sur un jeux de données manuellement annoté et constitué d'entités de départ et de voisins à conserver ou à élaguer. Même si les expériences et les cas d'utilisation ont mis en évidence la capacité de généralisation de notre modèle, il est possible que la définition apprise des décisions de conservation ou d'élagage d'entités voisines ne soit pas adaptée à d'autres applications. Pour répondre à cette question, nous prévoyons de

12. `https://www.wikidata.org/wiki/Q30335959`.

permettre aux utilisateurs de fournir leurs propres exemples de voisins conservés et élagués. Ces exemples pourraient être utilisés dans la phase d'inférence ou même pour entraîner à la volée des modèles personnalisés, étant donné la complexité réduite de nos modèles. Pour guider ces développements futurs, nous sommes à l'écoute des retours d'utilisation, cas d'usage, contextes applicatifs, et thèmes d'intérêt de la communauté.

Remerciements

Ces travaux sont soutenus par le projet AT2TA[13] (ANR-22-CE23-0023) financé par l'Agence nationale de la recherche (ANR).

Références

[1] Tim Berners-Lee, James Hendler, and Ora Lassila. 2001. The semantic web. *Scientific american* 284, 5: 28–37.

[2] Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems 26: 27th annual conference on neural information processing systems 2013. Proceedings of a meeting held december 5-8, 2013, lake tahoe, nevada, united states*, 2787–2795. Retrieved from https://proceedings.neurips.cc/paper/2013/hash/1cecc7a77928ca8133fa24680a88d2f9-Abstract.html.

[3] Jiaoyan Chen, Hang Dong, Janna Hastings, Ernesto Jiménez-Ruiz, Vanessa López, Pierre Monnin, Catia Pesquita, Petr Skoda, and Valentina A. M. Tamma. 2023. Knowledge graphs for the life sciences: Recent developments, challenges and opportunities. *Transactions on Graph Data and Knowledge* 1, 1: 5:1–5:33. https://doi.org/10.4230/TGDK.1.1.5.

[4] Claudia d'Amato, Louis Mahon, Pierre Monnin, and Giorgos Stamou. 2023. Machine learning and knowledge graphs: Existing gaps and future research challenges. *Transactions on Graph Data and Knowledge* 1, 1: 8:1–8:35. https://doi.org/10.4230/TGDK.1.1.8.

[5] Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. 2014. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *The 20th ACM SIGKDD international conference on knowledge discovery and data mining, KDD '14, new york, NY, USA - august 24 - 27, 2014*, 601–610. https://doi.org/10.1145/2623330.2623623.

[6] Michel Dumontier, Alison Callahan, Jose Cruz-Toledo, Peter Ansell, Vincent Emonet, François Belleau, and Arnaud Droit. 2014. Bio2RDF release 3: A larger, more connected network of linked data for the life sciences. In *Proceedings of the ISWC 2014 posters & demonstrations track a track within the 13th international semantic web conference, ISWC 2014, riva del garda, italy, october 21, 2014* (CEUR workshop proceedings), 401–404. Retrieved from https://ceur-ws.org/Vol-1272/paper_121.pdf.

[7] Mariano Fernández-López, Asuncion Gomez-Perez, and Natalia Juristo. 1997. METHONTOLOGY: From ontological art towards ontological engineering. *Engineering Workshop on Ontological Engineering (AAAI97)*.

13. https://at2ta.loria.fr/.

[8] Eleanor E. Fink, Pedro A. Szekely, and Craig A. Knoblock. 2014. How linked open data can help in locating stolen or looted cultural property. In *Digital heritage. Progress in cultural heritaage: Documentation, preservation, and protection - 5th international conference, EuroMed 2014, limassol, cyprus, november 3-8, 2014. proceedings* (Lecture notes in computer science), 228–237. https://doi.org/10.1007/978-3-319-13695-0_22.

[9] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard de Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, Axel-Cyrille Ngonga Ngomo, Axel Polleres, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan Sequeda, Steffen Staab, and Antoine Zimmermann. 2021. *Knowledge graphs*. Morgan & Claypool Publishers. https://doi.org/10.2200/S01125ED1V01Y202109DSK022.

[10] Lucas Jarnac, Miguel Couceiro, and Pierre Monnin. 2023. Relevant entity selection: Knowledge graph bootstrapping via zero-shot analogical pruning. In *Proceedings of the 32nd ACM international conference on information and knowledge management, CIKM 2023, birmingham, united kingdom, october 21-25, 2023*, 934–944. https://doi.org/10.1145/3583780.3615030.

[11] Lucas Jarnac and Pierre Monnin. 2022. Wikidata to bootstrap an enterprise knowledge graph: How to stay on topic? In *Proceedings of the 3rd wikidata workshop 2022 co-located with the 21st international semantic web conference (ISWC2022), virtual event, hanghzou, china, october 2022* (CEUR workshop proceedings). Retrieved from https://ceur-ws.org/Vol-3262/paper16.pdf.

[12] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Christian Bizer. 2015. DBpedia - A large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web* 6, 2: 167–195. https://doi.org/10.3233/SW-140134.

[13] Suryani Lim, Henri Prade, and Gilles Richard. 2019. Solving word analogies: A machine learning perspective. In *Symbolic and quantitative approaches to reasoning with uncertainty, 15th european conference, ECSQARU 2019, belgrade, serbia, september 18-20, 2019, proceedings* (Lecture notes in computer science), 238–250. https://doi.org/10.1007/978-3-030-29765-7_20.

[14] Jixiong Liu, Yoan Chabot, Raphaël Troncy, Viet-Phi Huynh, Thomas Labbé, and Pierre Monnin. 2023. From tabular data to knowledge graphs: A survey of semantic table interpretation tasks and methods. *Journal of Web Semantics* 76: 100761. https://doi.org/10.1016/J.WEBSEM.2022.100761.

[15] Farzaneh Mahdisoltani, Joanna Biega, and Fabian M. Suchanek. 2015. YAGO3: A knowledge base from multilingual wikipedias. In *Seventh biennial conference on innovative data systems research, CIDR 2015, asilomar, CA, USA, january 4-7, 2015, online proceedings*. Retrieved from http://cidrdb.org/cidr2015/Papers/CIDR15_Paper1.pdf.

[16] Laurent Miclet, Sabri Bayoudh, and Arnaud Delhay. 2008. Analogical dissimilarity: Definition, algorithms and two experiments in machine learning. *Journal of Artificial Intelligence Research* 32: 793–824. https://doi.org/10.1613/jair.2519.

[17] Melanie Mitchell. 2021. Abstraction and analogy-making in artificial intelligence. *Annals of the New York Academy of Sciences* 1505, 1: 79–101.

[18] Natalya Fridman Noy, Yuqing Gao, Anshu Jain, Anant Narayanan, Alan Patterson, and Jamie Taylor. 2019. Industry-scale knowledge graphs: Lessons and challenges. *Communications of the ACM* 62, 8: 36–43. https://doi.org/10.1145/3331166.

[19] Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. 2024. Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering*: 1–20. https://doi.org/10.1109/tkde.2024.3352100.

[20] Juan Sequeda and Ora Lassila. 2021. *Designing and building enterprise knowledge graphs*. Morgan & Claypool Publishers. https://doi.org/10.2200/S01105ED1V01Y202105DSK020.

[21] Basel Shbita, Anna Lisa Gentile, Pengyuan Li, Chad DeLuca, and Guang-Jie Ren. 2023. Understanding customer requirements - an enterprise knowledge graph approach. In *The semantic web - 20th international conference, ESWC 2023, hersonissos, crete, greece, may 28 - june 1, 2023, proceedings* (Lecture notes in computer science), 625–643. https://doi.org/10.1007/978-3-031-33455-9_37.

[22] Bill Swartout, Ramesh Patil, Kevin Knight, and Tom Russ. 1996. Toward distributed use of large-scale ontologies. In *Proceedings of the tenth workshop on knowledge acquisition for knowledge-based systems*, 25.

[23] Thomas Pellissier Tanon, Gerhard Weikum, and Fabian M. Suchanek. 2020. YAGO 4: A reason-able knowledge base. In *The semantic web - 17th international conference, ESWC 2020, heraklion, crete, greece, may 31-june 4, 2020, proceedings* (Lecture notes in computer science), 583–596. https://doi.org/10.1007/978-3-030-49461-2_34.

[24] Denny Vrandecic and Markus Krötzsch. 2014. Wikidata: A free collaborative knowledgebase. *Communications of the ACM* 57, 10: 78–85. https://doi.org/10.1145/2629489.

[25] Gerhard Weikum, Xin Luna Dong, Simon Razniewski, and Fabian M. Suchanek. 2021. Machine knowledge: Creation and curation of comprehensive knowledge bases. *Foundations and Trends Databases* 10, 2-4: 108–490.

[26] Laurel Zuckerman. 2020. Linked data and holocaust era art markets: Gaps and dysfunctions in the knowledge supply chain. In *Proceedings of the international conference collect and connect: Archives and collections in a digital age, leiden, the netherlands, november 23-24, 2020* (CEUR workshop proceedings), 13–24. Retrieved from https://ceur-ws.org/Vol-2810/paper2.pdf.

[27] Laurel Zuckerman. Tracking looted art with graphs: A case study. Retrieved from https://api.semanticscholar.org/CorpusID:247314664.