Evaluating Class Membership Relations in Knowledge Graphs using Large Language Models

Bradley P. Allen[0000−0003−0216−3930] and Paul T. Groth[0000−0003−0183−6910]

University of Amsterdam, Amsterdam, The Netherlands
{b.p.allen,p.t.groth}@uva.nl

Abstract. A backbone of knowledge graphs are their class membership relations, which assign entities to a given class. As part of the knowledge engineering process, we propose a new method for evaluating the quality of these relations by processing descriptions of a given entity and class using a zero-shot chain-of-thought classifier that uses a natural language intensional definition of a class. We evaluate the method using two publicly available knowledge graphs, Wikidata and CaLiGraph, and 7 large language models. Using the gpt-4-0125-preview large language model, the method's classification performance achieves a macro-averaged F1-score of 0.830 on data from Wikidata and 0.893 on data from CaLiGraph. Moreover, a manual analysis of the classification errors shows that 40.9% of errors were due to the knowledge graphs, with 16.0% due to missing relations and 24.9% due to incorrectly asserted relations. These results show how large language models can assist knowledge engineers in the process of knowledge graph refinement. The code and data are available on Github[1].

Keywords: Knowledge engineering, large language models, knowledge graph refinement, natural language generation

1 Introduction

Knowledge graphs (KGs) have become a key technology in many applications in industry and academia [20]. This has brought attention to the area of KG refinement [28], for which a main goal is ensuring that the knowledge captured in KGs is as complete and correct as possible. This is a challenge, given that large-scale KGs composed of contributions from multiple sources of knowledge often contain incomplete, misaligned, and inaccurate information [31,29]. At the same time, as part of the knowledge engineering process, direct manual evaluation of KG quality by human reviewers to detect and remediate these problems is expensive [37,18].

The recent emergence of large language models (LLMs) has inspired work towards understanding how LLMs can applied to knowledge graph construction.

[1] https://github.com/bradleypallen/evaluating-kg-class-memberships-using-llms

To date, much of this work has centered on the use of LLMs for knowledge graph completion [38] and the evaluation of provenance [5] and correctness [32] in a knowledge graph. In this paper, we describe work on using LLMs to evaluate *class membership relations* in a KG. Class membership relations are important because they are a principal way in which knowledge graphs represent classification schemes. Classification schemes are a major consideration in many knowledge engineering efforts, often with significant implications for social policy and scientific consensus [9].

We present an approach to evaluate class membership relations by using an LLM to define a zero-shot chain-of-thought (CoT) [23,36] classifier that takes natural language descriptions of an entity and a class in a given KG, and predicts whether or not the entity is an instance of the class, providing a natural language rationale for the prediction. The motivation for this approach is to leverage an LLM's capabilities for natural language processing to allow knowledge engineers to use intensional knowledge expressed in natural language by domain experts directly, as opposed to having to first transform it into a symbolic knowledge representation, and apply it to determining if that knowledge is accurately reflected in a given knowledge graph.

2 Related work

Using LLMs for knowledge engineering tasks Beyond uses for KG refinement, LLMs are beginning to be applied to other tasks in the engineering of knowledge graphs. In [4], two scenarios for the use of LLMs in knowledge engineering are described: creating hybrid neurosymbolic knowledge systems and enabling knowledge engineering in natural language. Pan et al [27] describe three categories of LLM/KG hybrids: KG-enhanced LLMs, LLM-augmented KGs, and synergized LLMs + KGs. Specific examples of LLM augmentation of KGs include the use of LLMs for KG completion [38,3] and for ontology engineering [26]. We view our work as an example of an LLM-augmented KG approach that performs knowledge engineering using intensional knowledge expressed in natural language to develop classifiers; classification is a well-known instance of an analytic knowledge task as defined in the CommonKADS taxonomy of knowledge-intensive task types [30].

KG refinement Knowledge graph refinement is defined by Paulheim [28] as the process of improving an existing KG by adding missing knowledge or identifying and removing errors. KG refinement has been implemented using manual, statistical, rule-based and hybrid methods [37,18]. Interactive solutions to aid human reviewers have been developed, including tools for crowdsourcing KG quality assessment [24], fact-checking triples using textual evidence [32], ontology repair using description logic reasoners [25], and sampling techniques to better focus manual reviewers' attention [13]. Our work builds on these results by creating classifiers that can be used to alert a knowledge engineer to misalignments between natural language definitions of a class and elements of the class's extension in a given KG.

Automated fact checking A recent survey [14] provides a useful overview of the large amount of methods for fact checking. The work most related to ours is that of Atanasova et al. [7] on justification production using language models of the BERT family. That work focuses on the fact checking applied to claims expressed as natural language statements; in contrast, our methods admit of the combination of both serialized RDF statements and natural language descriptions as input for both justification production and verdict prediction.

3 Preliminaries

To precisely specify the integration between KGs and LLMs in our experiments, we now introduce a formalization of a neurosymbolic workflow [12] for entity classification.

Language models Let $\mathcal{T}$ be the set of sequences of tokens $T_i = t_1, t_2, \ldots, t_n$ such that t_i is a token in a predefined vocabulary V. Given a *corpus* $\mathcal{C} \subseteq \mathcal{T}$, a *language model* $\mathcal{L}_\mathcal{C}$ is a probabilistic model trained on a sample of $\mathcal{C}$ that defines a distribution over sequences of tokens.

$$\mathcal{L}_\mathcal{C}(T_i) = p(t_1, t_2, \ldots, t_n) \tag{1}$$

is an estimate of the probability of a sequence T_i, given a corpus $\mathcal{C}$. A *prompt* $P = (T, F)$ is a pair of a sequence of tokens T and an set of *free* tokens $F \subseteq \{f_1, f_2, \ldots, f_n\}$. A *substitution* θ with respect to a prompt P is a set of pairs (f_i, T_i) such that $f_i \in F$ and $T_i \in \mathcal{T}$. An *instantiation* $\texttt{instantiate}(P, \theta)$ is a prompt P' such that $\forall (f_i, T_i) \in \theta$ every occurrence of f_i in P is replaced with f_i. Given a prompt P, the goal of a language model $\mathcal{L}_\mathcal{C}$ is to generate a sequence of tokens that maximizes the conditional probability under $\mathcal{L}_\mathcal{C}$.

$$T_{\text{out}} = \arg\max_T \mathcal{L}_\mathcal{C}(T|P) \tag{2}$$

is the output sequence generated by the language model, conditioned on P.

Knowledge graphs Following [6], we use the RDF data model to describe knowledge graphs. Let I be an infinite set of IRIs (Internationalized Resource Identifiers [11]), B be an infinite set of blank nodes [19], and L an infinite set of literals [8]. A *knowledge graph* G is a set of *triples* $\{(s, p, o) \mid s \in S, p \in P, o \in O\}$, where $S \subset I \cup B$ is the set of *subjects* in G, $P \subset I$ is the set of *properties* in G, and $O \subset I \cup B \cup L$ is the set of *objects* in G. Let $\texttt{instanceOf}, \texttt{subClassOf}, \texttt{label} \in P$ denote an instance-of relation, a subclass-of relation, and a label property in G, respectively. A *class* $c \in I \cup B$ is an entity that represents a set of entities sharing common properties and relationships in G. Let

$$\texttt{ext}(c) = \bigcup_{i \in \mathbb{N}} \texttt{ext}_i(c) \tag{3}$$

be the *extension* of a class c, where

$$\texttt{ext}_0(c) = \{e \mid \exists (e, \texttt{instanceOf}, c) \in G\} \tag{4}$$

$$\texttt{ext}_{i+1}(c) = \texttt{ext}_i(c) \cup \{e \mid e \in \texttt{ext}(c') \wedge \exists (c', \texttt{subClassOf}, c) \in G\} \tag{5}$$

Zero-shot chain-of-thought entity classifiers Given the definitions above, we now proceed to show how to construct classifiers that prompt LLMs with intensional definitions of classes in natural language to classify entities in a knowledge graph. For any entity $e \in I \cup B$, let $Ge = \{(s, p, o) \in G \mid s = e \vee o = e\}$ be the *neighborhood* of e. Let $T_{label(e)} = \{o \mid \exists(e, \texttt{label}, o) \in G\ \}$. A *serialization* T_G of a knowledge graph G is a sequence of tokens T that represents the triples in G using a structured formal language (e.g. RDF). For any entity $e \in E$, let T_{Ge} be the serialization of Ge. A *verbalization* T_e of an entity e is a sequence of tokens T that represents a description of e in natural language. Given an language model $\mathcal{L}_\mathcal{C}$, we define a function `classify` as follows:

$$(T_R, T_\mathbb{B}) = \texttt{classify}(c, e) \tag{6}$$

where T_R is a sequence of tokens that represents a rationale for a classification decision, and $T_\mathbb{B} \in \{\texttt{positive}, \texttt{negative}\}$ are tokens that represent classification decisions, i.e., whether or not $e \in \texttt{ext}(c)$, respectively. We instantiate T_R and $T_\mathbb{B}$ as follows:

$$T_R = \arg\max_T \mathcal{L}_\mathcal{C}(T|\texttt{instantiate}(P_{rationale\_generation}, \theta_0)) \tag{7}$$

$$T_\mathbb{B} = \arg\max_T \mathcal{L}_\mathcal{C}(T|\texttt{instantiate}(P_{answer\_generation}, \theta_1)) \tag{8}$$

$$\begin{aligned}\theta_0 = \{&(\{\texttt{label}\}, T_{label(c)}), (\{\texttt{definition}\}, T_c),\\ &(\{\texttt{entity}\}, T_{label(e)}), (\{\texttt{description}\}, T_e)\}\end{aligned} \tag{9}$$

$$\theta_1 = \theta_0 \cup \{(\{\texttt{rationale}\}, T_{R_e})\} \tag{10}$$

given two prompt templates $P_{rationale\_generation}$ and $P_{answer\_generation}$. The specific prompt templates used in the experiments were manually authored and iteratively refined between June 2023 and October 2023. Figure 3 shows an example of such a classifier instantiated for a class and entity in the CaLiGraph KG.

4 Experiments

To understand the potential of classifiers built using the above approach for the problem of KG refinement, we conducted experiments to explore two research questions:

Q_1: Can the classifiers exhibit good alignment with KGs? Much of the work on LLM/KG synergy to date is predicated on the idea that KGs, as curated sources of knowledge, can be used to address gaps in the knowledge obtainable from LLMs, or mitigate the problem of hallucination by grounding LLMs. This makes assumptions about the degree of alignment between LLMs and KGs, hence this question aims to measure this alignment.

Q_2: Can the classifiers detect missing or incorrect relations? Our main goal is to generate classifications based on intensional class definitions in with natural language rationales to help guide human reviewers to areas where KGs may be incomplete or incorrect. Any such approach must demonstrate the ability to do so across multiple knowledge graphs and classes.

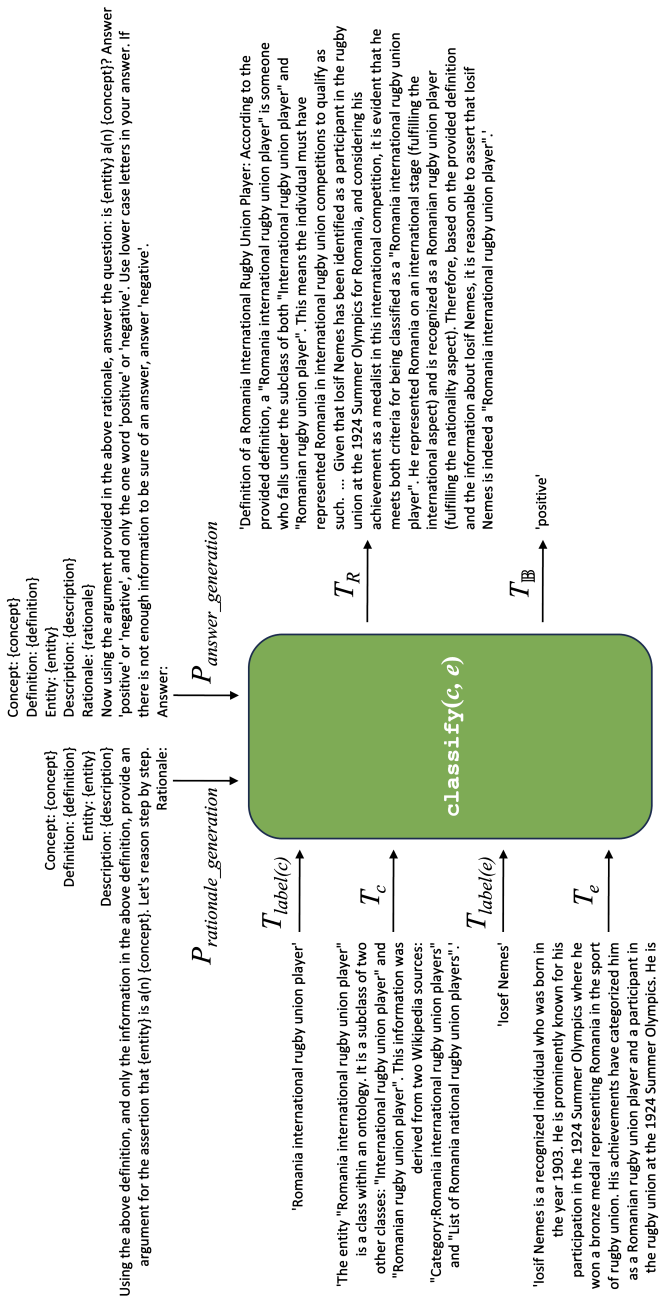

Fig. 1. A zero-shot chain-of-thought classifier applied to the class `clgo:Romania_international_rugby_union_player` and the entity `clgr:Iosif_Nemes` from the CaLiGraph knowledge graph [15].

The experiments to address these questions were implemented as follows:

Knowledge graphs Two publicly available knowledge graphs were used to construct evaluation datasets: Wikidata [34] and CaLiGraph [15,16,17]. The two KGs represent distinct approaches to KG construction. Wikidata is the result of the crowd-sourced contribution of factual statements by thousands of human contributors and automated processes working independently, yielding relatively diverse approaches to modeling concepts and entities, and is loosely coupled with and derived from information in Wikipedia. CaLiGraph is the result of the automated extraction of terminology and assertions from Wikipedia and DBPedia pages, and as such is more consistent in how it models concepts and entities than is Wikidata.
Data sets We randomly sampled 20 classes from Wikidata and 19 from CaLiGraph using SPARQL queries, including their super-classes. For each, 20 entities were selected as positive examples, and up to 20 entities from the set difference of the extensions of the class and one of its superclasses as negative examples (in some of the sampled classes the cardinality of the set difference was less than 20).
Language models We evaluated seven large language models accessible using services provided by OpenAI and Hugging Face: OpenAI's gpt-4-0125-preview and gpt-3.5-turbo, Google's gemma-7b-it and gemma-2b-it, Mistral AI's Mixtral-8x7B-Instruct-v0.1 and Mistral-7B-Instruct-v0.2, and Meta's Llama-2-70b-chat-hf. For all experiments, temperatures were set to a value of 0.1.
Classifiers We use the definitions provided above to instantiate a classifier for each class in the datasets. For each class c and entity e, we obtained natural language descriptions to use as T_c and T_e arguments for the classifier. For Wikidata, we retrieved natural language summaries of the class or entity from its associated Wikipedia page. For CaLiGraph, we used gpt-4-1106-preview to generate RDF verbalizations to serve as T_c and T_e, given inputs of T_{G_c} and T_{G_e} obtained as the TSV serialization of the triples returned from SPARQL DESCRIBE queries for c and e with LIMIT = 20.
Evaluation procedure Experimental runs were then conducted by applying classifiers to each class/entity pair for a given class in each of the two knowledge graphs, generating for each class a confusion matrix based on the resulting set of classifications, from which performance metrics are computed. Algorithm 1 describes this procedure in pseudo-code. Evaluations whose statistics are reported below were conducted during the period from 24 February 2024 to 27 February 2024. Costs incurred through calls to language model APIs during this period totalled around $225 USD.

5 Findings

We summarize below the finding obtained from our evaluations. Detailed results can be found in our aforementioned Github repository.

Classifier performance (assuming the KG as ground truth) of the seven closed- and open-source LLMs is shown in Table 1. The performance as shown

input : a pair of classes c, d from $G \mid (c, \texttt{subClassOf}, d) \in G$
output: a confusion matrix M

$(TP, FP, TN, FN) \leftarrow (0, 0, 0, 0)$;
$E^+ \leftarrow$ a sample from $\texttt{ext}(c)$;
$E^- \leftarrow$ a sample from $\texttt{ext}(d) \setminus \texttt{ext}(c)$;
foreach $e \in E^+$ **do**
  $(T_R, T_{\mathbb{B}}) \leftarrow \texttt{classify}(c, e)$;
  if $T_{\mathbb{B}} = \textit{positive}$ **then** $TP \leftarrow TP + 1$;
  else $FP \leftarrow FP + 1$;
end
foreach $e \in E^-$ **do**
  $(T_R, T_{\mathbb{B}}) \leftarrow \texttt{classify}(c, e)$;
  if $T_{\mathbb{B}} = \textit{negative}$ **then** $TN \leftarrow TN + 1$;
  else $FN \leftarrow FN + 1$;
end
$M \leftarrow [[TP, FP], [FN, TN]]$;

Algorithm 1: Evaluation procedure

supports the following finding: **classifiers can exhibit good alignment with KGs** (Q_1). As evidenced by Cohen's κ values, one LLM was in moderate agreement with Wikidata, and four were in moderate agreement with CaLiGraph.

KG	LLM	ACC	AUC	F1	κ
Wikidata	gpt-4-0125-preview	**0.830**	**0.830**	**0.823**	**0.660**
	gemma-7b-it	0.726	0.727	0.705	0.454
	Mixtral-8x7B-Instruct-v0.1	0.697	0.696	0.654	0.393
	Mistral-7B-Instruct-v0.2	0.671	0.671	0.620	0.342
	gemma-2b-it	0.674	0.670	0.629	0.330
	gpt-3.5-turbo	0.627	0.627	0.547	0.255
	Llama-2-70b-chat-hf	0.631	0.616	0.569	0.239
CaLiGraph	gpt-4-0125-preview	**0.900**	**0.893**	**0.889**	**0.788**
	Mixtral-8x7B-Instruct-v0.1	0.893	0.884	0.874	0.767
	gpt-3.5-turbo	0.842	0.833	0.815	0.665
	Mistral-7B-Instruct-v0.2	0.812	0.803	0.779	0.605
	gemma-7b-it	0.783	0.774	0.750	0.547
	Llama-2-70b-chat-hf	0.637	0.625	0.558	0.252
	gemma-2b-it	0.563	0.543	0.422	0.090

Table 1: Classifier performance by LLM.

Table 2 shows the results of an error analysis of the evaluation results for the highest-performing classifier (using gpt-4-0125-preview). It was conducted by having one of the authors manually annotate each classification error with their own classification decision, based on the information in the provided descriptions. This human judgment was then compared with that of the KG and classifier using the pairwise Cohen's κ value as a measure of inter-annotator

agreement. In cases where where Wikidata and the classifier using gpt-4-0125-preview disagreed, the human showed fair agreement with Wikidata and no agreement with the classifier, and for examples where CaLiGraph and the given classifier disagreed, the human showed slight agreement with the classifier and no agreement with CaLiGraph.

In addition, the annotator assigned each error to one of five causes: missing data in the entity description that comprised the LLM's ability to classify the entity, a missing class membership relation in the KG between the given entity and class (an example of which is shown in Figure 3), an incorrectly asserted class membership relation in the KG between the given entity and class, and an error on the part of the LLM, through either hallucination or misinterpretation of the class definition or entity description. We assert that these results support

KG	N	FP	FN	human-KG κ	human-LLM κ	missing data	missing relation	incorrect relation	incorrect reasoning
Wikidata	136	46	90	**0.243**	-0.241	34 (25.0%)	15 (11.0%)	33 (24.3%)	54 (39.7%)
CaLiGraph	77	27	50	-0.295	**0.198**	28 (36.4%)	19 (24.7%)	20 (26.0%)	10 (13.0%)
	213	73	140			62 (29.1%)	34 (16.0%)	53 (24.9%)	64 (30.0%)

Table 2: Summary of the analysis of classification errors by gpt-4-0125-preview.

another finding: **classifiers can detect missing or incorrect relations** in KGs (Q_2). The error analysis showed that in instances where the classifier using gpt-4-0125-preview was in disagreement with the KG, 40.9% of errors were due to the knowledge graphs, with 16.0% due to missing relations and 24.9% due to incorrect relations. 29.1% of the errors could be ascribed to missing or insufficient data in the entity description, which may have had a negative impact on classifier performance. This is attributed primarily to one of two reasons: for CaLiGraph, RDF verbalizations missed relevant information about entities due to the omission of relevant triples in the set produced by the SPARQL DESCRIBE queries; and for Wikidata, some entities had descriptions that were simply the label assigned to the entity. We plan to address these shortcomings in future versions of the evaluation datasets. These results suggest that, *pâce* efforts focused on using KGs to mitigate knowledge gaps and hallucinations in LLMs, LLMs may have a corresponding role to play in mitigating knowledge gaps and errors in KGs.

6 Discussion

Contributions The principal contributions of this work are 1) a formal approach to the design of a neurosymbolic knowledge engineering workflow integrating KGs and LLMs, and 2) experimental evidence that this method can assist knowledge engineers in addressing the correctness and completeness of KGs, potentially reducing the effort involved in knowledge acquisition and elicitation.

Limitations Challenges with the use of LLMs include the cost of API calls to proprietary LLMs and the speed of processing tasks with such resource-intensive systems. Our results show the potential for open source, locally deployed LLMs to address the first problem; we expect that sampling approaches, frequently used in other approaches to KG refinement in large-scale KGs, can help address the second. The human evaluation for error analysis could be improved through the use of crowd-sourcing to expand the number of reviewers (allowing much larger sets of rationales and classification decisions to be evaluated), by evaluating the true positives and true negatives produced by the classifier, and by evaluating the soundness of rationales and faithfulness of classification to the given rationales. The potential impact of one or more of the LLMs having processed the Wikidata and CaLiGraph data during pre-training was not considered in the analysis. The question of whether the use of gpt-4-1106-preview to generate RDF verbalizations in the CaLiGraph experiments approach to verbalization introduced bias relative to the other LLMs is yet to be addressed. Finally, this work is limited to the evaluation of class membership relations in a KG, and evaluated against KGs that are domain-general and either crowdsourced (Wikidata) or automatically generated from crowdsourced content (CaLiGraph). To support use against KG refinement challenges faced by domain-specific KGs, such as those developed for life sciences applications [10], this needs to be generalized to support the definition of classifiers based on intensional definitions of predicates in natural language.

Future work We have in this work taken a minimalist approach to the prompt engineering of classifiers, restricting ourselves to a zero-shot chain-of-thought approach. Expanding this to include using temperature sampling [1] for self-consistency [35] and uncertainty estimation [21], mitigating hallucination in rationale generation [22], and addressing faithfulness in rationale generation [33,2] are three other areas for future work, in addition to work on addressing the limitations described above by expanding the number and types of relations considered, and evaluating our approach against domain-specific KGs.

Acknowledgements

This work is partially funded by the Dutch Research Council (NWO) through grant MVI.19.032. The authors wish to thank Filip Ilievski, Jan-Christoph Kalo, Xue Li, Fina Polat, Thivyan Thanapalasingam, and Lise Stork for discussions and suggestions that have been invaluable in refining this work. We would also like to thank the anonymous reviewers for their insightful comments and suggestions, which have been invaluable in refining our work.

References

1. Ackley, D.H., Hinton, G.E., Sejnowski, T.J.: A learning algorithm for boltzmann machines. Cognitive Science **9**(1), 147–169 (1985). https://doi.org/10.7551/mitpress/4943.003.0039, `http://dx.doi.org/10.7551/mitpress/4943.003.0039`

2. Agarwal, C., Tanneru, S.H., Lakkaraju, H.: Faithfulness vs. plausibility: On the (un) reliability of explanations from large language models. arXiv preprint arXiv:2402.04614 (2024)
3. Alivanistos, D., Santamaría, S.B., Cochez, M., Kalo, J.C., van Krieken, E., Thanapalasingam, T.: Prompting as probing: Using language models for knowledge base construction. arXiv preprint arXiv:2208.11057 (2022)
4. Allen, B.P., Stork, L., Groth, P.: Knowledge engineering using large language models. Transactions on Graph Data and Knowledge **1**(1), 3:1–3:19 (2023). https://doi.org/10.4230/TGDK.1.1.3, `https://drops.dagstuhl.de/entities/document/10.4230/TGDK.1.1.3`
5. Amaral, G., Rodrigues, O., Simperl, E.: Prove: A pipeline for automated provenance verification of knowledge graphs against textual sources. arXiv preprint arXiv:2210.14846 (2022)
6. Angles, R., Thakkar, H., Tomaszuk, D.: Mapping rdf databases to property graph databases. IEEE Access **8**, 86091–86110 (2020)
7. Atanasova, P., Simonsen, J.G., Lioma, C., Augenstein, I.: Generating fact checking explanations. arXiv preprint arXiv:2004.05773 (2020)
8. Beek, W., Ilievski, F., Debattista, J., Schlobach, S., Wielemaker, J.: Literally better: Analyzing and improving the quality of literals. Semantic Web **9**(1), 131–150 (2018)
9. Bowker, G.C., Star, S.L.: Sorting things out: Classification and its consequences. MIT press (2000)
10. Chen, J., Dong, H., Hastings, J., Jiménez-Ruiz, E., López, V., Monnin, P., Pesquita, C., Škoda, P., Tamma, V.: Knowledge graphs for the life sciences: Recent developments, challenges and opportunities. Transactions on Graph Data and Knowledge **1**(1), 5:1–5:33 (2023). https://doi.org/10.4230/TGDK.1.1.5, `https://drops.dagstuhl.de/entities/document/10.4230/TGDK.1.1.5`
11. Dürst, M., Suignard, M.: Internationalized resource identifiers (iris). Tech. rep., RFC Editor (2005)
12. Ekaputra, F.J., Llugiqi, M., Sabou, M., Ekelhart, A., Paulheim, H., Breit, A., Revenko, A., Waltersdorfer, L., Farfar, K.E., Auer, S.: Describing and organizing semantic web and machine learning systems in the swemls-kg. In: European Semantic Web Conference. pp. 372–389. Springer (2023)
13. Gao, J., Li, X., Xu, Y.E., Sisman, B., Dong, X.L., Yang, J.: Efficient knowledge graph accuracy evaluation. arXiv preprint arXiv:1907.09657 (2019)
14. Guo, Z., Schlichtkrull, M., Vlachos, A.: A survey on automated fact-checking. Transactions of the Association for Computational Linguistics **10**, 178–206 (2022)
15. Heist, N., Paulheim, H.: The caligraph ontology as a challenge for owl reasoners. arXiv preprint arXiv:2110.05028 (2021)
16. Heist, N., Paulheim, H.: Information extraction from co-occurring similar entities. In: Proceedings of the Web Conference 2021. pp. 3999–4009 (2021)
17. Heist, N., Paulheim, H.: Transformer-based subject entity detection in wikipedia listings. arXiv preprint arXiv:2210.01482 (2022)
18. Hofer, M., Obraczka, D., Saeedi, A., Köpcke, H., Rahm, E.: Construction of knowledge graphs: State and challenges. arXiv preprint arXiv:2302.11509 (2023)
19. Hogan, A., Arenas, M., Mallea, A., Polleres, A.: Everything you always wanted to know about blank nodes. Journal of Web Semantics **27**, 42–69 (2014)
20. Hogan, A., Blomqvist, E., Cochez, M., D'amato, C., Melo, G.D., Gutierrez, C., Kirrane, S., Gayo, J.E.L., Navigli, R., Neumaier, S., Ngomo, A.C.N., Polleres, A., Rashid, S.M., Rula, A., Schmelzeisen, L., Sequeda, J., Staab, S.,

Zimmermann, A.: Knowledge graphs. ACM Comput. Surv. **54**(4) (jul 2021). https://doi.org/10.1145/3447772, https://doi.org/10.1145/3447772
21. Huang, Y., Song, J., Wang, Z., Chen, H., Ma, L.: Look before you leap: An exploratory study of uncertainty measurement for large language models. arXiv preprint arXiv:2307.10236 (2023)
22. Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y.J., Madotto, A., Fung, P.: Survey of hallucination in natural language generation. ACM Computing Surveys **55**(12), 1–38 (2023)
23. Kojima, T., Gu, S.S., Reid, M., Matsuo, Y., Iwasawa, Y.: Large language models are zero-shot reasoners. Advances in neural information processing systems **35**, 22199–22213 (2022)
24. Kontokostas, D., Zaveri, A., Auer, S., Lehmann, J.: Triplecheckmate: A tool for crowdsourcing the quality assessment of linked data. In: Knowledge Engineering and the Semantic Web: 4th International Conference, KESW 2013, St. Petersburg, Russia, October 7-9, 2013. Proceedings 4. pp. 265–272. Springer (2013)
25. Lambrix, P.: Completing and debugging ontologies: State of the art and challenges in repairing ontologies. ACM Journal of Data and Information Quality (2023)
26. Mateiu, P., Groza, A.: Ontology engineering with large language models. arXiv preprint arXiv:2307.16699 (2023)
27. Pan, S., Luo, L., Wang, Y., Chen, C., Wang, J., Wu, X.: Unifying large language models and knowledge graphs: A roadmap. arXiv preprint arXiv:2306.08302 (2023)
28. Paulheim, H.: Knowledge graph refinement: A survey of approaches and evaluation methods. Semantic web **8**(3), 489–508 (2017)
29. Piscopo, A., Simperl, E.: What we talk about when we talk about wikidata quality: a literature survey. In: Proceedings of the 15th International Symposium on Open Collaboration. pp. 1–11 (2019)
30. Schreiber, A.T., Schreiber, G., Akkermans, H., Anjewierden, A., Shadbolt, N., de Hoog, R., Van de Velde, W., Wielinga, B.: Knowledge engineering and management: the CommonKADS methodology. MIT press (2000)
31. Shenoy, K., Ilievski, F., Garijo, D., Schwabe, D., Szekely, P.: A study of the quality of wikidata. Journal of Web Semantics **72**, 100679 (2022)
32. Syed, Z.H., Röder, M., Ngonga Ngomo, A.C.: Factcheck: Validating rdf triples using textual evidence. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management. pp. 1599–1602 (2018)
33. Turpin, M., Michael, J., Perez, E., Bowman, S.R.: Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. arXiv preprint arXiv:2305.04388 (2023)
34. Vrandečić, D., Krötzsch, M.: Wikidata: a free collaborative knowledgebase. Communications of the ACM **57**(10), 78–85 (2014)
35. Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., Narang, S., Chowdhery, A., Zhou, D.: Self-consistency improves chain of thought reasoning in language models. arXiv preprint arXiv:2203.11171 (2022)
36. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q.V., Zhou, D., et al.: Chain-of-thought prompting elicits reasoning in large language models. Advances in Neural Information Processing Systems **35**, 24824–24837 (2022)
37. Xue, B., Zou, L.: Knowledge graph quality management: a comprehensive survey. IEEE Transactions on Knowledge and Data Engineering (2022)
38. Zhang, B., Reklos, I., Jain, N., Peñuela, A.M., Simperl, E.: Using large language models for knowledge engineering (llmke): A case study on wikidata. arXiv preprint arXiv:2309.08491 (2023)