Wiki Workshop (10th edition) – May 11, 2023

Wikidata to Bootstrap an Enterprise Knowledge Graph: How to Stay on Topic?

Lucas Jarnac
Orange
Belfort, France
lucas.jarnac@orange.com

Pierre Monnin
Orange
Belfort, France
pierre.monnin@orange.com

Abstract

In this work, we propose to bootstrap an Enterprise Knowledge Graph with Wikidata by automatically selecting parts of its hierarchy related to business domains of interest. In particular, given seed QIDs of interest, we perform an expansion along the hierarchy and prune unrelated QIDs with degree and embedding distance-based thresholds. Our results show that distance in the embedding space is an effective pruning feature but node degree is still necessary.

Keywords: Knowledge Graph, Pruning, Node Degree, Graph Embedding, Distance

Introduction

Enterprise Knowledge Graphs (EKGs) are major assets of companies since they support various downstream applications such as knowledge sharing, data integration, search, or question answering (Noy and others, 2019). Building an EKG is an iterative and continuous process that can be carried out with automatic knowledge extraction approaches from semi-structured or textual data. In such a view, it is common to first build a high quality KG nucleus with entities and categories extracted from premium sources. This nucleus will then support automatic knowledge extraction systems applied on a wider variety of data sources (Weikum and others, 2021).

In our work, we propose to build an EKG nucleus from Wikidata (Vrandecic and Krötzsch, 2014). Specifically, we perform an expansion along the ontology hierarchy starting from given business terms mapped to Wikidata entities (Figure 1). However, since Wikidata contains numerous classes, there is a need to limit this expansion by pruning classes unrelated to the business topics of interest. To this aim, we propose to rely on distance between node embeddings and node degree. This paper is an extended abstract of (Jarnac and Monnin, 2022).

Methods

To prune unrelated classes when traversing the hierarchy of Wikidata classes from seed business terms, we rely on node degree and distance in the embedding space.

Node degree. The degree of a node is defined as the sum of its incoming and outgoing edges. In our case, we compute the degree of classes by considering incoming and outgoing P31 and P279 edges since they are the only edges traversed in our expansion. We assume that the degree of a class is representative of its generality. Hence, classes with a high degree may be to generic and may deviate from the original business domains.

Distance in the embedding space. Since classes can have instances, we propose the two following definitions for the distance between a class and the starting QID:

Definition 1 (Distance $\mathcal{D}_1$). *The distance between a class and the starting QID is the Euclidean distance between their embeddings.*

Definition 2 (Distance $\mathcal{D}_2$). *The distance between a class and the starting QID is the Euclidean distance between the centroids of the embeddings of their respective instances. In case the class or the starting QID has no instance, its embedding is used instead.*

We assume that distances in the embedding space represent topic relatedness between classes. Hence, classes with a high distance between their embeddings and the embedding of the starting QID may deviate from the original business domains.

Our expansion and pruning algorithm is outlined in Algorithm 1 and makes use of three following thresholds:

Absolute degree threshold $\tau_{\textbf{degree-abs}}(c)$. This threshold prunes classes whose degree is greater than an input parameter.

Relative degree threshold $\tau_{\textbf{degree-rel}}(c)$. The absolute degree threshold may not always be applicable. Consider classes reached at a specific expansion level. It is possible for some of them to have much higher degrees than the other classes of the same level without these degrees being pruned by $\tau_{\text{degree-abs}}$. Such classes are regarded as anomalies, and thus pruned with an approach commonly used in anomaly detection.

At each expansion level, we compute the first (Q_1) and third (Q_3) quartiles of the degree of the traversed classes. We prune those whose degree is greater than $Q_3 + \alpha \times (Q_3 - Q_1)$, where α is an input parameter that controls how much class degree is allowed to deviate.

Algorithm 1 Expansion and pruning algorithm

```
Input: A set of seed QIDs $\mathcal{S}$
Output: A set of kept QIDs of interest $\mathcal{K}$
  $\mathcal{K} \leftarrow \emptyset$
  for $q \in \mathcal{S}$ do
    $Q \leftarrow \text{directClasses}(q)$
    while $Q \neq \emptyset$ do
      $\mathcal{N} \leftarrow \emptyset$
      for $c \in Q$ do
        if $\tau_{\text{degree-abs}}(c)$ and $\tau_{\text{degree-rel}}(c)$ and $\tau_{\text{dist-rel}}(c)$
        then
          $\mathcal{K} \leftarrow \mathcal{K} \cup \{c\}$
          $\mathcal{N} \leftarrow \mathcal{N} \cup \text{superClasses}(c) \cup \text{subClasses(c)}$
        end if
      end for
      $Q \leftarrow \mathcal{N}$
    end while
  end for
```

We compute this relative threshold at each expansion level since we assume degree may vary depending on the level but should be consistent at a given level. We apply this threshold if and only if the maximum degree at an expansion level exceeds a parameter γ. This allows to retrieve all classes from an expansion level if they all have a low degree, regardless of discrepancies in their degrees.

Relative distance threshold $\tau_{\text{dist-rel}}(c)$. This threshold prunes classes whose distance with the starting QID is greater than $\beta \times \mu_{\text{dist-cl}}$. β is an input coefficient that controls the allowed range of distances and $\mu_{\text{dist-cl}}$ is the mean distance between the starting QID and its direct classes cl. We assume that these direct classes are closely related to the starting QID and can serve as a basis to measure the remoteness of other classes in the embedding space. Note that direct classes whose distance with the starting QID is greater than the third quartile of distances plus the interquartile range are removed. These classes are abnormally far from the starting QID, and thus may not constitute a correct basis for remoteness.

Results

We experimented our approach with the pre-trained embeddings of Wikidata available in PyTorch-BigGraph (Lerer and others, 2019). 839 business terms were semi-manually matched with their corresponding Wikidata entities before applying the expansion (Figure 1). Their expansion retrieved 393 distinct direct classes, 946 distinct super-classes, and 2,560,426 distinct sub-classes. From these results, we chose to focus only on pruning sub-classes since their important number may indicate unrelated classes to the original business terms. We performed six expansion and pruning experiments from the 839 original business terms with different configurations (~ 5 minutes per expansion with pruning). We fixed $\tau_{\text{degree-abs}} = 200$, $\alpha = 1.5$, $\gamma = 20$ and tested with $\beta \in \{1.2, 1.25, 1.3\}$ and the two distances $\mathcal{D}_1$ and $\mathcal{D}_2$. We manually labeled the pruned and kept classes to evaluate the performance of our approach. Results are presented in Table 1. Each row presents the results of one experiment in which all defined thresholds are applied.

Discussion & Conclusion

Table 1 shows that $\mathcal{D}_2$ obtains the best global pruning precision. Additionally, $\mathcal{D}_2$ obtains a better precision than $\mathcal{D}_1$ for distance-based pruning. This result was expected as it appears through our experiments that a distance based on centroids of class instances better carries the relatedness between classes (Figure 2). Interestingly, more classes are pruned and kept with $\mathcal{D}_2$ than with $\mathcal{D}_1$, which indicates that $\mathcal{D}_2$ leads to a different and more extensive hierarchical exploration than $\mathcal{D}_1$. Overall, our experiments show that distance in the embedding space is a promising feature for pruning, especially when considering the embedding of a class as the centroid of the embeddings of its instances. However, results also highlight that node degree is an effective and necessary feature that cannot be substituted by distance in the embedding space since some classes are only pruned by degree thresholds. In future works, we ambition to confirm these results with different graph embedding models, investigate the learning of graph embeddings specific to the pruning task, and compare with feature-based similarity measures and propagation-based graph metrics.

References

[Jarnac and Monnin2022] 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 ISWC2022*.

[Lerer and others2019] Adam Lerer et al. 2019. Pytorch-biggraph: A large scale graph embedding system. In *Proceedings of Machine Learning and Systems 2019*. mlsys.org.

[Noy and others2019] Natalya Fridman Noy et al. 2019. Industry-scale knowledge graphs: lessons and challenges. *Communications of the ACM*, 62(8):36–43.

[Vrandecic and Krötzsch2014] Denny Vrandecic and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

[Weikum and others2021] Gerhard Weikum et al. 2021. Machine knowledge: Creation and curation of comprehensive knowledge bases. *Foundations and Trends Databases*, 10(2-4):108–490.

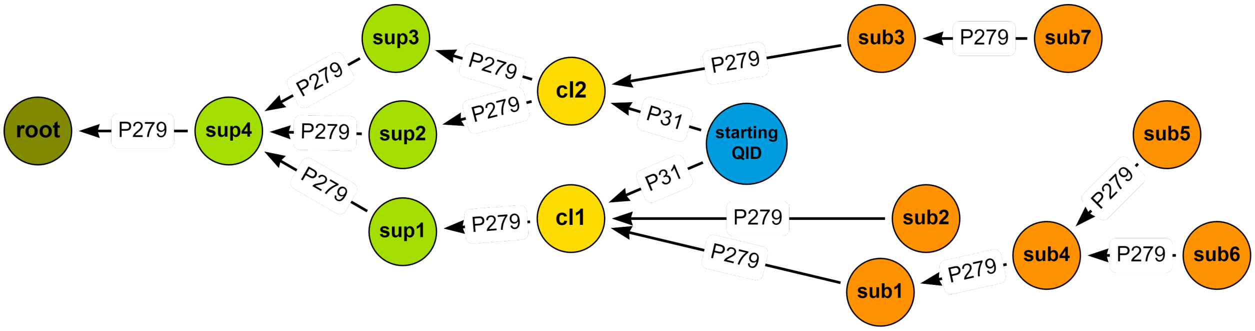

Figure 1: Illustration of the expansion along the ontology hierarchy starting from an original business term mapped to a Wikidata entity (starting QID). We first retrieve its direct classes following P31 edges (cl_i), and then their super-classes (sup_j) and sub-classes (sub_k) following P279 edges.

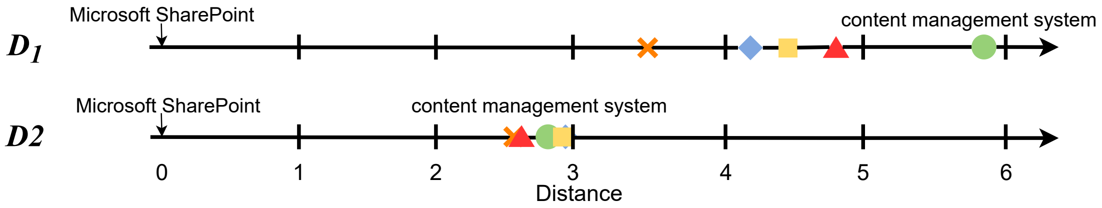

Figure 2: Distances between the original business term "Microsoft SharePoint" and its direct classes (expansion level 1): social software (✕), document management system (◆), Enterprise Content Management system (🟨), server software (▲), and content management system (●). With $\mathcal{D}_1$, direct classes are dispersed. For example, "content management system" is far from the starting QID. On the contrary, $\mathcal{D}_2$ leads to a better organization of distances.

Table 1: Number and precision of pruned and kept classes with the two distances and different configurations for β. (1) indicates classes pruned with $\tau_{\text{degree-rel}}$; (2) indicates classes pruned with $\tau_{\text{degree-abs}}$; (3) indicates classes pruned with $\tau_{\text{dist-rel}}$; (4) indicates classes pruned with both $\tau_{\text{degree-rel}}$ and $\tau_{\text{dist-rel}}$; (5) indicates global pruning. Values in bold indicate the best precision.

	β	# Pruned classes					Precision					# Kept classes	Precision
		(1)	(2)	(3)	(4)	(5)	(1)	(2)	(3)	(4)	(5)		
$\mathcal{D}_1$	1.2	11	45	1,344	182	1,582	**0.91**	**0.87**	0.69	0.78	0.71	1,135	**0.86**
	1.25	19	46	1,289	183	1,537	0.89	**0.87**	0.73	0.79	0.74	1,293	0.83
	1.3	25	46	1,224	189	1,484	0.88	**0.87**	0.75	0.79	0.76	1,484	0.77
$\mathcal{D}_2$	1.2	184	60	2,108	118	2,470	0.74	0.85	0.75	0.92	0.76	1,645	0.81
	1.25	213	64	2,032	110	2,419	0.77	0.84	0.79	0.92	0.79	1,931	0.76
	1.3	250	65	1,917	100	2,332	0.79	0.83	**0.84**	**0.94**	**0.84**	2,311	0.71