Unsupervised Hierarchical Grouping of Knowledge Graph Entities

Sameh K. Mohamed[1,2,3]

[1] Data Science Institute
[2] Insight Centre for Data Analytics
[3] National University of Ireland Galway
sameh.mohamed@insight-centre.org

Abstract. Knowledge graphs have attracted lots of attention in academic and industrial environments. Despite their usefulness, popular knowledge graphs suffer from incompleteness of information especially in their type assertions. This has encouraged research in the automatic discovery of entity types. In this context, multiple works were developed to utilise logical inference on ontologies and statistical machine learning methods to learn type assertion in knowledge graphs. However, these approaches suffer from limited performance on noisy data, limited scalability and the dependence on labelled training samples. In this work, we propose a new unsupervised approach that learns to categorise entities into a hierarchy of named groups. We show that our approach is able to effectively learn entity groups using a scalable procedure in noisy and sparse datasets. We experiment our approach on a set of popular knowledge graph benchmarking datasets, and we publish a collection of the outcome group hierarchies[4].

Keywords: Knowledge Graphs · Entity Clustering · Heirarical Clustering.

1 Introduction

Type information of knowledge graph (KG) entities is an important feature that categorizes entities of similar semantics. It is used in different tasks related to knowledge graphs including meta path extraction [1], link prediction [2] and fact checking [3]. Naturally, knowledge entities are categorized into a hierarchically structured set of classes *e.g.*, *person* ⇒ *artist* ⇒ *singer*, or *location* ⇒ *country* ⇒ *city*, where each class encloses entities of similar properties. Hierarchical class structure provides richer semantics of entity type information that can be used as a feature in different knowledge graph tasks *e.g.* link prediction and entity linking [4]. Despite their usefulness, famous knowledge graphs from different domains suffer from type assertion incompleteness [5]. For example, type assertions of DBpedia 3.8 is estimated to have at most an upper bound of completeness 63.7% [6]. YAGO2 types are also estimated to be at most 53.3%

[4] https://samehkamaleldin.github.io/kg-hierarchies-gallery/

complete [6]. This incompleteness has motivated research into automatic discovery of knowledge graph entity types.

The currently available approaches for classifying knowledge graph entities can be classified into two categories. First, schema-based approaches, where the developed approaches utilise the known schema of the knowledge graph to learn entity types. This includes automatically inferring type information using standard RDF reasoning [7], which applies rules of logical inference to infer new type assertion statements from existing ones using knowledge graph schema information. However, this approach is sensitive to noisy information, and it depends on prior-defined ontologies, and its predictions are bounded by the set of types defined by the ontology. The SDType [8] model is another schema-based approach which introduced using the link based type inference approach which depends on the assumption that relations happen between particular types. For example, if entities e_1 and e_2 are connected with relation "*LiveIn*", then we can infer that e_1 is a person and e_2 is a place using relation defined domain and range types from schema. This approach provided efficient classification of entities in noisy knowledge graphs. However, it depended on the presence of schema information, and it also did not provide hierarchical structure of type information.

Secondly, the statistical learning approaches, which treat type prediction task as a multi-label classification problem, where models learn entity types using a set of graph based features. They use known type assertions as training examples and learn a model that can infer other unknown type assertions. They are known to provide more robust-to-noise type predictions [6] compared to schema-based and logic-based approaches. These models can also infer type hierarchies by using hierarchical multi-label classifiers [5]. However, these models require training assertions and it can not suggest new entity types.

In this work, we propose a new unsupervised approach that groups knowledge graph entities according to their connected neighbour in the graph. This can be formulated such that for every set of entity nodes E connected to another object node e_o with a relation r, the set of nodes E belongs to the group g_{r,e_o}. For example, in a general knowledge graph about people and cities all the people entities connected to the entity "*Dublin*" with the relation "*live-in*" are associated with the group "*live-in-Dublin*" as shown in Fig. 1. We then learn the hierarchy of these groups using intersection and containment ratios between them to build a hierarchy of entity types. This allows us to produce new named groups for entities, and provides rich hierarchy of the newly created groups.

The rest of this work is organised as follows:

(A) We present a detailed description of the pipeline of our approach in Section 2.

(B) We experiment our method on multiple popular knowledge graph benchmarking datasets and we show samples of produced hierarchies in Section 3.

(C) We discuss related works in Section 4.

(D) We present our conclusions and future works in Section 5.

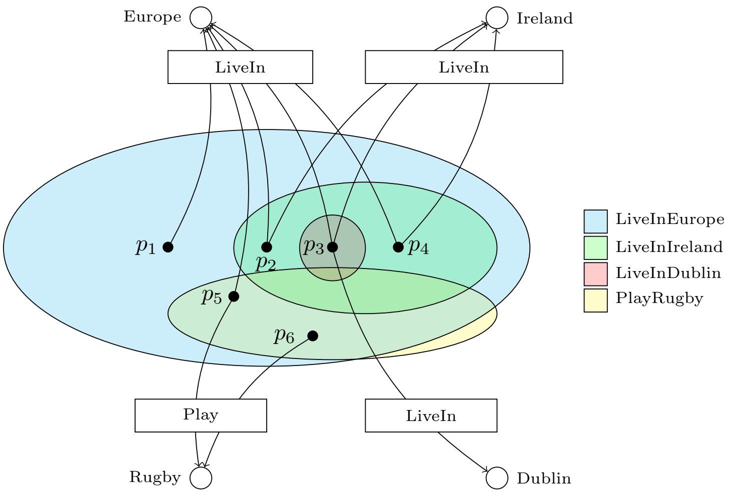

Fig. 1. Venn diagram of groups of people from a sample of facts about people living in Europe

2 Methods

In this section, we present our approach for learning hierarchical groups of entities in knowledge graphs. We divide the pipeline of our method into three segments: (1) entity grouping, (2) computing group similarities and (3) building group hierarchy. In the following subsection, we discuss the motivation for our approach and we discuss also each of its pipeline segments in detail while we use the knowledge graph sample in Fig. 1 as a running example.

2.1 Motivation

Knowledge graphs can model different types of assertions depending on their predicate types such as attribute assertions like the age or names of an entity, or related entities like birthplaces, friend, etc. Usually, the cardinality of relational predicates are many-to-many like having a friend, associated nationality or working for an organisation. In this case, our brains intuitively group entities associated with same predicates and destinations to a group like "friend of Jack", "people with British nationality " or "companies working in IT". In our approach, we use the same technique, where we transform every knowledge assertion *i.e.* SPO triple to a group assertion ("S" belongs to group "PO"). We also use a configurable minimum size requirement for groups to avoid including one-to-one predicates as group assertions.

Algorithm 1 Generating entity groups in a knowledge graph

```
Require: group min. size $\alpha$, KG triplets $T$, num. of jobs $j$, group dictionary $\mathcal{D}$
initialise $d_1, d2, ..., d_j$ as empty dictionaries
$t_1, t_2, ..., t_j \leftarrow$ split$(T, j)$
parfor $i \leftarrow [1, 2, ..., j]$ do
    for $(s, p, o) \in t_i$ do
        group $\leftarrow$ concatenate$(p, o)$
        if group in $d_i$ then
            $d_i$[group].append$(s)$
        else
            $d_i$[group] $\leftarrow [s]$
        end if
    end for
end parfor
for $i \leftarrow [1, 2, ..., j]$ do
    $\mathcal{D} \leftarrow \mathcal{D} + d_i$
end for
for $g \in \mathcal{D}$ do
    if size$(g) < \alpha$ then delete $\mathcal{D}[g]$
    end if
end for
return $D$
```

2.2 Generating Entity Groups

Given any knowledge graph, our approach starts with generating groups of entities by transforming the graph facts into group assertions as previously discussed. For example, the knowledge graph in Fig. 1 contains multiple facts about six persons. In this graph, all the persons are living in Europe such that $\forall p \in \{p_1, ..., p_6\}(p, ''LiveIn'', ''Europe'')$. This is transformed to creating a group called "*LiveIn_ Europe*" (g) such that $g = \{p_1, p_2, p_3, p_4, p_5, p_6\}$. Similarly, other groups are created like "*LiveIn_ Ireland*"$\mapsto \{p_2, p_3, p_4\}$, "*LiveIn_ Dublin*" $\mapsto \{p_3\}$ and "*Play_ Rugby*"$\mapsto \{p_5, p_6\}$. This procedure is performed to generate groups from all the triples in the knowledge graph.

Algorithm 1 describes the process of generating these groups, where processing facts is parallelised to speed up processing large volumes of data. First, the set of all knowledge graph triples is divided into a configurable j number of splits. Each of these splits is then processed to generate a dictionary of groups and their contained entities according to their own set of triples. All the resulting dictionaries are then joined to generate one dictionary of all groups in the graph and their corresponding entities. In order to restrict the extracted groups to a minimum specific number of entities, groups with size less than a configurable minimum size are removed from the group dictionary. Finally, the outcome of this procedure is a dictionary of the remaining groups and their member entities.

2.3 Computing Group Similarity

We compute similarity measures between entity groups to learn their similarities and hierarchical structure. We compute two types of similarities to achieve that: Jaccard and hub promoted index (HPI) similarities [9] which can be defined as follows:

$$S_{g1,g2}^{jaccard} = \frac{\mathit{\Gamma}(g1) \cap \mathit{\Gamma}(g2)}{\mathit{\Gamma}(g1) \cup \mathit{\Gamma}(g2)} \quad , \quad S_{g1,g2}^{HPI} = \frac{|\mathit{\Gamma}(g1) \cap \mathit{\Gamma}(g2)|}{\min(|g1|, |g2|)},$$

for any two groups $g1$ and $g2$, where $\mathit{\Gamma}(g)$ denotes the set of member entities of the group g and $|g|$ denotes its size. The HPI similarity in this context computes the overlap between two groups, where its maximum value 1 implies that one of the groups is a subset of the other. The Jaccard similarity on the other hand computes the overall similarity between two groups of entities. For example, the HPI similarity between "*LiveInEurope*" and "*LiveInIreland*" is equal to $|\{p2, p3, p4\}|/min(3, 6) = 1$, which implies that the small group "*LiveInIreland*" is a subset of the larger group "*LiveInEurope*".

We also compute the similarities in a parallel procedure similar to the generation of groups. We first generate all possible combinations of groups, then we divide these combinations into splits. We then compute similarities for each of the splits. the outcome similarities of all the parallel procedures are then joined to generate a similarity matrix between all the groups.

2.4 Building Groups Hierarchy

The range of the hub promoted index (HPI) similarity between two groups is bounded between 0 and 1, where 0 represents that the groups are independent and 1 implies that one group is a subset of the other. In noisy knowledge graph, the HPI index between totally dependant groups is always less than 1 due to missing members in one of the two groups. In our approach we use a configurable parameter θ that represents the group containment HPI threshold. We initialise this parameter with a value of 0.9 by default to tolerate 10% of information loss.

3 Experiments

In this section, we describe the datasets and outcomes of the experimentation of our approach.

3.1 Data

In our experiments we use six knowledge graph benchmarking datasets:

- WN18 & WN18RR: subsets of the WordNet dataset [10] which contain lexical information of the English language [11,12].
- FB13k: a subset of the freebase dataset [13] which contains information about general human knowledge [11,14].

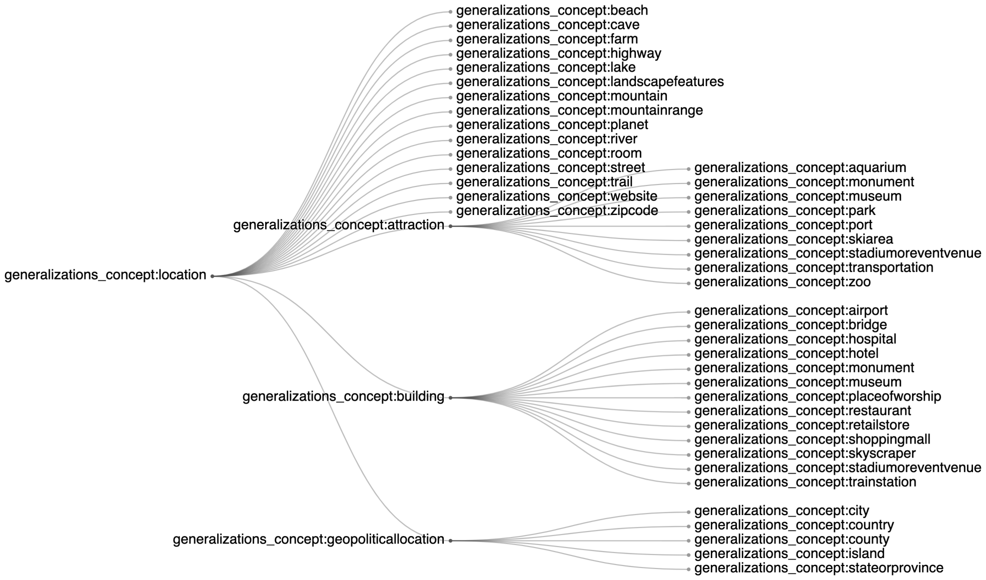
generalizations_concept:location
generalizations_concept:attraction
generalizations_concept:beach
generalizations_concept:cave
generalizations_concept:farm
generalizations_concept:highway
generalizations_concept:lake
generalizations_concept:landscapefeatures
generalizations_concept:mountain
generalizations_concept:mountainrange
generalizations_concept:planet
generalizations_concept:river
generalizations_concept:room
generalizations_concept:street
generalizations_concept:trail
generalizations_concept:website
generalizations_concept:zipcode
generalizations_concept:aquarium
generalizations_concept:monument
generalizations_concept:museum
generalizations_concept:park
generalizations_concept:port
generalizations_concept:skiarea
generalizations_concept:stadiumoreventvenue
generalizations_concept:transportation
generalizations_concept:zoo
generalizations_concept:building
generalizations_concept:airport
generalizations_concept:bridge
generalizations_concept:hospital
generalizations_concept:hotel
generalizations_concept:monument
generalizations_concept:museum
generalizations_concept:placeofworship
generalizations_concept:restaurant
generalizations_concept:retailstore
generalizations_concept:shoppingmall
generalizations_concept:skyscraper
generalizations_concept:stadiumoreventvenue
generalizations_concept:trainstation
generalizations_concept:geopoliticallocation
generalizations_concept:city
generalizations_concept:country
generalizations_concept:county
generalizations_concept:island
generalizations_concept:stateorprovince

Fig. 2. An example of extracted location-based group hierarchy of a set of entities from the NELL239 dataset.

- YAGO10: a subset of of YAGO3 dataset [15] which contains information mostly about people and their citizenship, gender, and profession knowledge [16].
- NELL239: a subset of NELL dataset [17,18] which contains general knowledge about people, places, sports teams, universities, etc.

The above mentioned datasets are divided into three splits: training, validation and testing. In our experiments, we first join the three splits and we execute our method on the full dataset.

3.2 Outcomes

We executed our approaches on the aforementioned datasets, and we have generated the entity group hierarchies for each one of them. We use a minimum group size of 10 and a threshold of 0.9 for the HPI similarity for all the experimented datasets.

The outcome results show that these datasets contain different root groups, where each root include super group of different group semantics such locations, people and organisations.

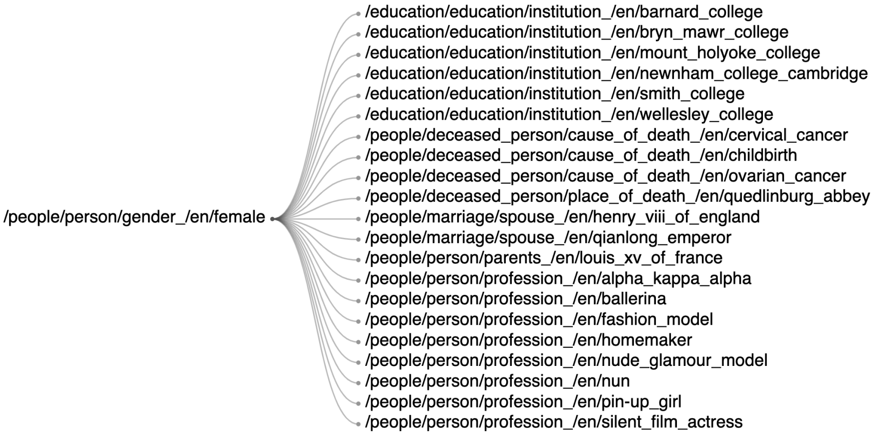
/people/person/gender_/en/female
/education/education/institution_/en/barnard_college
/education/education/institution_/en/bryn_mawr_college
/education/education/institution_/en/mount_holyoke_college
/education/education/institution_/en/newnham_college_cambridge
/education/education/institution_/en/smith_college
/education/education/institution_/en/wellesley_college
/people/deceased_person/cause_of_death_/en/cervical_cancer
/people/deceased_person/cause_of_death_/en/childbirth
/people/deceased_person/cause_of_death_/en/ovarian_cancer
/people/deceased_person/place_of_death_/en/quedlinburg_abbey
/people/marriage/spouse_/en/henry_viii_of_england
/people/marriage/spouse_/en/qianlong_emperor
/people/person/parents_/en/louis_xv_of_france
/people/person/profession_/en/alpha_kappa_alpha
/people/person/profession_/en/ballerina
/people/person/profession_/en/fashion_model
/people/person/profession_/en/homemaker
/people/person/profession_/en/nude_glamour_model
/people/person/profession_/en/nun
/people/person/profession_/en/pin-up_girl
/people/person/profession_/en/silent_film_actress

Fig. 3. An example of extracted gender-based group hierarchy of a set of entities for females from the FB13 dataset.

Fig. 2 shows an example of an outcome hierarchy of location-based entities in the NELL239 dataset. It shows that our approach is able to generate a hierarchy of different levels with valid and meaningful semantics.

Fig 3 also shows another hierarchy for entities of people with a female gender extracted from the FB13k dataset [19]. The results show a one level hierarchy where these entities are categorised into multiple groups. For example, the group of those having an education in the Barnard College is a subset of the group of people with a female gender. We have investigated on the Barnard college and found out that it is a private women's liberal arts college located in Manhattan, New York City. Founded in 1889 by Annie Nathan Meyer, who named it after Columbia University's 10th president, Frederick Barnard, it is one of the oldest women's colleges in the world. Similarly for other associated colleges in Fig. 3, these colleges are all women colleges.

We have also published further outcomes and hierarchy views on a publicly available website [5].

[5] https://samehkamaleldin.github.io/kg-hierarchies-gallery/

3.3 Implementation

The experiments are implemented in Python3.5, and all the experiments were executed on a Linux machine with an Intel(R) Core(TM) i70.4790K CPU @ 4.00GHz and 32 GB RAM. We also use the D3 JavaScript library to visualise our hierarchy in a radial tree form.

4 Related Work

Despite the widespread uses of knowledge graph in multiple domains, they suffer from missing information, especially type-based assertions of their entities. Multiple works were developed to tackle this problem including classical link prediction models [20], where multiple model use graph features [21] and embeddings [22] to learn type links in the knowledge graphs. Also, type assertions of entities can be learnt using association rule mining models [23,24] that identify type-based rules in the graph.

Further works have also focused on developing methods that exclusively predict entity types in knowledge graphs. These models have utilised different techniques including schema-based inference of type information [7] and combination of ontologies and graph patterns [8]. Furthermore, other techniques have utilised machine learning models to infer types where they learn a feature representation of entities and their known types [6,5], and use these learnt features to infer new type link for other untyped entities in the knowledge graph.

5 Conclusions and Future Work

In this work, we have discussed the problem on knowledge graph entity classification, and we have shown that current state-of-the-art solutions are limited. We have also proposed a new approach for hierarchical grouping for knowledge graph entities which utilises an intuitive grouping of entities connected with the same predicate object combinations. We have shown that our approach can provide named hierarchical categorisation for the knowledge graph entities in a scalable parallel procedure. Our approach also operates on noisy data data by using flexible similarity measures. We have experimented our approach on standard knowledge graph benchmarking datasets and we have published the outcome hierarchies.

In future works we intend to add the new generated groups as triples to the graph along with their equivalence and dependence relationships and evaluate their effects on tasks like link prediction on knowledge graphs. We also intend to examine the possible use of the outcome hierarchies in mining association rules in knowledge graphs.

6 Acknowledgements

This work has been supported by Insight Centre for Data Analytics at National University of Ireland Galway, Ireland (supported by the Science Foundation Ireland grant 12/RC/2289).

References

1. Changping Meng, Reynold Cheng, Silviu Maniu, Pierre Senellart, and Wangda Zhang. Discovering meta-paths in large heterogeneous information networks. In *WWW*, pages 754–764. ACM, 2015.
2. Denis Krompaß, Stephan Baier, and Volker Tresp. Type-constrained representation learning in knowledge graphs. In *International Semantic Web Conference (1)*, volume 9366 of *Lecture Notes in Computer Science*, pages 640–655. Springer, 2015.
3. Baoxu Shi and Tim Weninger. Discriminative predicate path mining for fact checking in knowledge graphs. *Knowl.-Based Syst.*, 104:123–133, 2016.
4. Ruobing Xie, Zhiyuan Liu, and Maosong Sun. Representation learning of knowledge graphs with hierarchical types. In *IJCAI*, pages 2965–2971. IJCAI/AAAI Press, 2016.
5. André Melo, Heiko Paulheim, and Johanna Völker. Type prediction in RDF knowledge bases using hierarchical multilabel classification. In *WIMS*, pages 14:1–14:10. ACM, 2016.
6. Heiko Paulheim and Christian Bizer. Improving the quality of linked data using statistical distributions. *Int. J. Semantic Web Inf. Syst.*, 10(2):63–86, 2014.
7. Axel Polleres, Aidan Hogan, Renaud Delbru, and Jürgen Umbrich. RDFS and OWL reasoning for linked data. In *Reasoning Web*, volume 8067 of *Lecture Notes in Computer Science*, pages 91–149. Springer, 2013.
8. Heiko Paulheim and Christian Bizer. Type inference on noisy RDF data. In *International Semantic Web Conference (1)*, volume 8218 of *Lecture Notes in Computer Science*, pages 510–525. Springer, 2013.
9. Linyuan Lu and Tao Zhou. Link prediction in complex networks: A survey. *CoRR*, abs/1010.0725, 2010.
10. George A. Miller. WordNet: A lexical database for english. *Commun. ACM*, 38(11):39–41, 1995.
11. Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *NIPS*, pages 2787–2795, 2013.
12. Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2d knowledge graph embeddings. In *AAAI*. AAAI Press, 2018.
13. Kurt D. Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *SIGMOD Conference*, pages 1247–1250. ACM, 2008.
14. Kristina Toutanova, Danqi Chen, Patrick Pantel, Hoifung Poon, Pallavi Choudhury, and Michael Gamon. Representing text for joint embedding of text and knowledge bases. In *EMNLP*, pages 1499–1509. The Association for Computational Linguistics, 2015.
15. Farzaneh Mahdisoltani, Joanna Biega, and Fabian M. Suchanek. YAGO3: A knowledge base from multilingual wikipedias. In *CIDR*. www.cidrdb.org, 2015.

16. Guillaume Bouchard, Sameer Singh, and Théo Trouillon. On approximate reasoning capabilities of low-rank vector spaces. In *AAAI Spring Syposium on Knowledge Representation and Reasoning (KRR): Integrating Symbolic and Neural Approaches*. AAAI Press, 2015.
17. Tom M. Mitchell, William W. Cohen, Estevam R. Hruschka Jr., Partha P. Talukdar, Bo Yang, Justin Betteridge, Andrew Carlson, Bhavana Dalvi Mishra, Matt Gardner, Bryan Kisiel, Jayant Krishnamurthy, Ni Lao, Kathryn Mazaitis, Thahir Mohamed, Ndapandula Nakashole, Emmanouil A. Platanios, Alan Ritter, Mehdi Samadi, Burr Settles, Richard C. Wang, Derry Wijaya, Abhinav Gupta, Xinlei Chen, Abulhair Saparov, Malcolm Greaves, and Joel Welling. Never-ending learning. *Commun. ACM*, 61(5):103–115, 2018.
18. Matt Gardner and Tom M. Mitchell. Efficient and expressive knowledge base completion using subgraph feature extraction. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 1488–1498, 2015.
19. Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 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.*, pages 2787–2795, 2013.
20. Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1):11–33, 2016.
21. Sameh K. Mohamed, Vít Novácek, and Pierre-Yves Vandenbussche. Knowledge base completion using distinct subgraph paths. In *SAC*, 2018.
22. Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. Knowledge graph embedding: A survey of approaches and applications. *IEEE Trans. Knowl. Data Eng.*, 29(12):2724–2743, 2017.
23. Luis Galárraga, Christina Teflioudi, Katja Hose, and Fabian M. Suchanek. Amie: association rule mining under incomplete evidence in ontological knowledge bases. In *WWW*, 2013.
24. Sameh K. Mohamed, Emir Muñoz, Vít Novácek, and Pierre-Yves Vandenbussche. Identifying equivalent relation paths in knowledge graphs. In *LDK*, 2017.