UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft
Faculty for Computer Science, Electrical Engineering and Mathematics
Department of Computer Science
Research Group Data Science

Seminar Report
Submitted to the Data Science Research Group
in Partial Fullfilment of the Requirements for the Degree of
Masters of Computer-Science

Link Prediction in Knowledge Graphs with Concepts of Nearest Neighbours

by
Nihal Yadalam Murali Kumar

Thesis Supervisor:
Prof. Dr. Axel Ngonga

Paderborn, June 15, 2020

Erklärung

Ich versichere, dass ich die Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen worden ist. Alle Ausführungen, die wörtlich oder sinngemäß übernommen worden sind, sind als solche gekennzeichnet.

Ort, Datum	Unterschrift

Abstract. Knowledge Graphs plays a paramount role in data querying and data extraction in day-to-day life. Link prediction or predicting missing edges has always been in the fore-front of research, in the field of machine learning. In this paper, author explains the rule-based and latent-based approaches used in link prediction and also their advantages and disadvantages. Rule-based approach is learned by a specific rule set and is often time consuming. Latent based approach can be challenging to interpret the triples and their relationships. Author introduces a simple, but an effective method for link prediction. This method is based on the usage of Concepts of Nearest Neighbours (CNN) supported by Dempster-Shafer theory. This approach does not require a training phase and the time required for predicting a link is effectively reduced. Experimental evaluation on the FB15k-237 data set using CNN shows, this approach is effective and it outperforms most of the state-of-the-art latent based tools.

Contents

1 Introduction 1

2 Related Work 5

3 Method 7

4 Results 11

5 Conclusion and Future work 13

Bibliography 14

1
Introduction

Advancement of software systems used in diverse domains led to the accumulation of data in large scale. Interpretation of the data extracted is essential to draw conclusions out of it. Data can be presented in many variations. Data sources helps in the construction of knowledge graphs which benefit several sectors of people around the world. Data can be of different types such as relational data i.e., relational databases, semi-structured data in the form of JSON, HTML etc, unstructured data i.e. images and documents. A lot of knowledge graphs makes use of knowledge databases like Wikipedia, Freebase. Knowledge extraction is achieved by extracting the knowledge by means of Natural Language Processing, text mining and machine learning techniques. Knowledge graphs provide semantically structured information which is understandable by computers. Knowledge graphs are made up of millions of entities or nodes and the edges or links between them. The main goal of entity extraction is to build relationships between entities. In this paper we concentrate on prediction of a link i.e., missing edges in a knowledge graph. Commonly used approaches in link prediction include latent based and rule based approaches. Latent based approaches are generally preferred in large-scale networks (knowledge graphs) for link predictions [NMTG15]. Rule-based approaches requires quiet an effort in computing inference between entities. This paper highlights the concept of finding patterns among entities using the concept of CNN. Dempster-Shafer theory supports in drawing conclusions from CNN. CNN saves fair amount of time in drawing inferences among entities by eliminating the training phase.

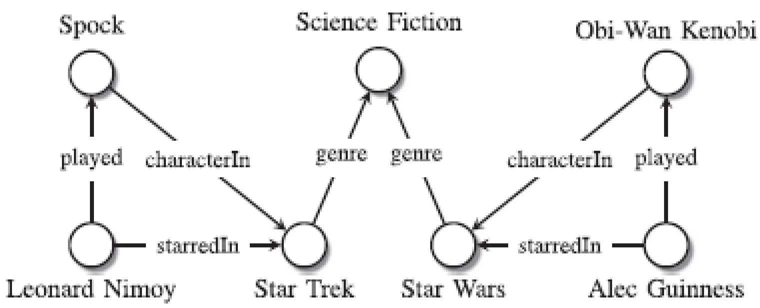

Figure 1.1: Example Knowledge Graph

Knowledge graph representation: KG represent knowledge in the form of entities and relationships between them. In fig 1.2, subject and object represent entities and predicate represent relationship. [NMTG15]

subject	*predicate*	*object*
(LeonardNimoy,	*profession,*	*Actor)*
(LeonardNimoy,	*starredIn,*	*StarTrek)*
(LeonardNimoy,	*played,*	*Spock)*
(Spock,	*characterIn,*	*StarTrek)*
(StarTrek,	*genre,*	*ScienceFiction)*

Figure 1.2: SPO triple representation

Open world and Closed world assumption in a Knowledge graph:

Consider fig 1.1 Nodes represent entities, edges represent types of relations. We define open world and closed world assumption [NMTG15]

Closed world Assumption:- In the CWA, tuples which are not present indicate false relationships. For example,in fig 1.1 there is no starredIn edge from Leonard Nimoy to Star Wars. It meant that Nimoy definitely did not star in this movie.

Open world Assumption:- In the OWA, tuples which are not present indicate as unknown, i.e., the respective relationship can be either true or false. For the same example, in fig 1.1 the missing edge is not interpreted to mean that Nimoy did not star in Star Wars.

Link prediction objective: Link prediction was first demonstrated in social networks for a binary or single relation [LNK07] and later introduced to data with multiple relations in [NMTG15]. We try to predict the link among its nearest neighbours by giving suitable rank to each of the nodes. Fewer the rank number, better is it's prediction. There exists few complexities in link prediction

1. KG's contain millions of links between entities which makes them difficult to classify
2. some links possess multivalued data.

Concept of KNN: Simple way to classify data

1. We place a new random element which can lie anywhere in the clusters of known elements.
2. We closely map the value of a random element to one of its neighbours say some value of k (k is the number of nearest neighbours) by some numerical distance also known as Manhattan distance.
3. Suppose the value of k = 1, then we map the value of a single nearest neighbour to this random element. The disadvantage is that it is more prone to be an outlier. To avoid this we take an optimum value of k (normally between 3 to 10) to classify the value of random element. On the contrary, large values of k leads to smoothing which might not be desirable for classification.

Formula for calculating Euclidean Distance in N dimensions for a KG:

$d_{man} = \sum_{i=1}^{N} |X_i - Y_i|$

- X , Y : Two data points
- N : N features per data
- X_i : ith feature of X

This approach is widely used in the classification problems in the field of machine learning

Advantages of KNN approach include:

1. No training phase needed which saves time.
2. Provides meaningful insight about each inferred link.
3. Provides good performance on link prediction compared to other state-of-the-art-tools.

Applications of Knowledge Graphs:

1. Knowledge graphs are used in Big Data applications in commercial and scientific domains.
2. Knowledge graphs aids in driving many intelligent machines [BKS13].
3. Knowledge graphs are used to build question answering system like IBM's question answering system.

2 Related Work

Latent feature models and Graph feature models are the two commonly used models used in link prediction [NMTG15].

1. **Latent Feature model:** Latent features are the features which are not directly observed in the data rather, it makes a fair assumption about inferences of the nodes. We define a statistical model for a knowledge graph [NMTG15]. Let $\varepsilon = \{e_1, ..., e_{Ne}\}$ represent the set of all entities and $R = \{r_1, ..., r_{Nr}\}$ represent the set of all relations in a knowledge graph. We consider triple of the form $x_{ijk} = (e_i, r_k, e_j)$ over this set of all entities, relations and a binary random variable of the form $y_{ijk} \in 0, 1$ which indicates its presence. The possible values of y_{ijk} can be as follows:

$$y_{ijk} = \begin{cases} 1 & \text{if the triple } (e_i, r_k, e_j) \text{ exists} \\ 0 & \text{otherwise} \end{cases} \tag{2.1}$$

   Consider an example, Alec Guinness received an academic award because, he is a good actor. The fact good actor is observed as a latent feature. We represent the latent feature of an entity e_i to a vector $e_i \in R_{H_e}$ where H_e denotes the count of the latent feature. The vectors are represented in the form

$$e_{Guinness} = \begin{bmatrix} 0.9 \\ 0.2 \end{bmatrix} e_{AcademyAward} = \begin{bmatrix} 0.2 \\ 0.8 \end{bmatrix} \tag{2.2}$$

   The component e_{i1} denotes the latent feature i.e., good actor and e_{i2} denotes the prestigious award. By looking at the values of these vectors, we can conclude these vectors are closely related to each other. TransE [BUGD+13] used the concept of latent-feature model and ranks a candidate tuple by the distance between the head and the tail. The data sets FB15K and WN18 from [BUGD+13] are used as the references for evaluation of link prediction for KNN. These data sets are taken from Wordnet and Freebase.

   The above data sets posed challenges due to the presence of inverse tuples i.e., change in one tuple falls on the opposite side of another. Improved data set FB15k-237 i.e,.removing all inverse tuples, was introduced by Toutanova and Chen [HFD12]. Later in this paper, we use this data set for evaluation.

2. **Graph Feature model:** In Graph feature model, [NMTG15] we predict the links or edges by extracting the features from the observed links in the graph, unlike latent feature model. For example, we could predict the parents of a person by a triple (John, married To, Mary) from the existence of the tuples (John, parentOf, Anne) and (Mary, parentOf, Anne) denoting a common child.

Both the feature models are good in their own way. However, they require an initial training phase which is time-consuming. KNN approach does not require training phase and they infer tuple relationships at a faster rate. This is demonstrated in the later phase of the paper.

3 Method

Mathematical Notations: A Knowledge graph is represented by $K = \langle E, R, T \rangle$, where $E \rightarrow$ Set of nodes, $R \rightarrow$ Set of relations and $T \subseteq E \times R \times T$ is the set of edges known as triples. Triple of the form (e_i, r_k, e_j) denotes a mapping or a relation r_k from a node e_i to a node e_j.

We represent a KG of the British royal family as a working example [Fer19]

$E = \{Charles, Diana, William, Harry, Kate, George, Charlotte, Louis, male, female\}$
$R = \{parent, spouse, sex\}$
$T = \{(\{William, Harry\}, parent, \{Charles, Diana\}),$
$(\{George, Charlotte, Louis\}, parent, \{William, Kate\}),$
$(Charles, spouse, Diana), (Diana, spouse, Charles),$
$(William, spouse, Kate), (Kate, spouse, William),$
$(\{Charles, William, Harry, George, Louis\}, sex, male),$
$(\{Diana, Kate, Charlotte\}, sex, female)\}$

The notation in the above knowledge graph of the form $(\{a, b\}, r, \{c, d\})$ is an abbreviation for $(a, r, c), (a, r, d), (b, r, c), (b, r, d)$. *Queries* related to graph patterns are used in CNNs and can be later used to draw inferences. There are two types of queries

1. **Triple Form:** The query of the form $(x, r, y) \in V \times R \times V$ is like a triple except nodes are replaced by variables.
2. **Boolean Form:** The query forms a Boolean expression on variables and nodes. We concentrate on the Boolean expression of the form $x = e$ i.e., a variable and a node.

In this paper, we use equality filters where nodes are represented in a triple form. The advantages include

- It is used to simplify the triple form to a $(Var - Var)$ form.
- Provides more flexibility for querying purpose i.e., more options in querying a KG.

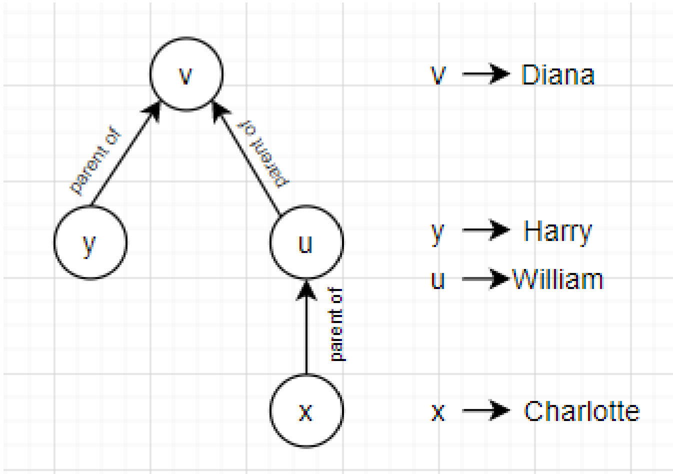

Figure 3.1: Sample Query example

Query: A Query is of the form $Q = (x_1, ..., x_n) \leftarrow P$ where P is called the projection of a graph on a subset of its variables. Consider fig 3.1,
$Q_{e,x} = (x, y) \leftarrow (x, parent, u), (u, parent, v), (y, parent, v), (y, sex, s), s = male$
gives us all (person, uncle) pairs. It implies that y is a male and also a sibling of a parent of x.

Matching or a Result set: A Matching or a Result Set is defined by a pattern P on a KG, $K = \langle E, R, T \rangle$ is a mapping μ from variables in P to nodes in E such that $\mu(t) \in T$ for each triple of the form $t \in P$. In the above example KG, the result set of the query is
$\mu_{ex} = \{x \mapsto Charlotte, y \mapsto Harry, u \mapsto William, v \mapsto Diana, s \mapsto male\}$

Concept of Nearest Neighbours (CNN): We define important terminologies to understand the concept of Nearest Neighbours. [Fer19]

1. **Graph Concept:** For the knowledge graph $K = \langle E, R, T \rangle$, a Graph concept is denoted as a pair $C = (A, Q)$, where A denotes a set of nodes and Q is a query such that $A = ans(Q)$ yields the set of answers related to Q and $Q = msq(A)$ yields the most specific query which verifies $A = ans(Q)$. A is called the extension $ext(C)$ of the Graph concept and Q is called the intension $int(C)$ of the Graph concept.
Consider a KG of British family, common inference between William and Charlotte is they have married parents. Query of the form:
$Q_{WC} = x \leftarrow (x, sex, s), (x, parent, y), (y, sex, t), t = male, (x, parent, z), (z, sex, u), u = female, (y, spouse, z), (z, spouse, y)$ from the definition of Graph concept, we have $A_{WC} = ans(Q_{WC})$ yields $William, Harry, George, Charlotte, Louis$ which match the definition $C_{WC} = (A_{WC}, Q_{WC})$.

2. **Conceptual distance:** Consider two nodes $n_1, n_2 \in E$. The conceptual distance between n_1 and n_2 is the most specific graph concept. It's extension contains both the entities i.e., $\delta(n_1, n_2) = (A, Q)$ infer $Q = msq(\{n_1, n_2\})$, $A = ans(Q)$. Intension Q denotes the common properties between two nodes in a result set i.e., $\cap$ and extension A denotes union of the nodes in a result set i.e., $\cup$.

l	S_l	$\|ext(\delta_l)\|$	$int(\delta_l)$	$\{l' \mid \delta_{l'} \lneq \delta_l\}$
1	$\{Charlotte\}$	1	$x \leftarrow x = Charlotte$	-
2	$\{Diana, Kate\}$	3	$x \leftarrow (x, sex, s), s = female$	1
3	$\{George, Louis\}$	3	$x \leftarrow (x, sex, s), (x, parent, y), y = William, \ldots$	1
4	$\{William, Harry\}$	5	$x \leftarrow (x, sex, s), (x, parent, y), \ldots$	1, 3
5	$\{Charles\}$	8	$x \leftarrow (x, sex, s)$	1, 2, 3, 4
6	$\{male, female\}$	10	$x \leftarrow \emptyset$	1, 2, 3, 4, 5

Figure 3.2: 6 Nearest Neighbours of Charlotte

Consider fig 3.2 [Fer19], There are 6 partitions in column 2 denoted by S_l and these partitions are made such that, the nodes present in the each partition are at the same conceptual distance from Charlotte. In the intension $int(\delta_l)$ column i.e., column 4, it contains several kinds of queries which result in the the S_l set. In the extension $\mid ext(\delta_l) \mid$ column i.e., column 3 indicates the numerical distance associated from each S_l set to Charlotte.

Algorithm and Aspects: The key concept of the algorithm is to loop through each of the S_l set of nodes repetitively and refine them such that, they result in proper partition of nodes. [Fer18]

Steps to be performed in KNN Algorithm:

- Each partition set S_l has a specific set of query element set associated to it and is denoted by H_l. When H_l is empty, there will be no further partition meaning that the result set is refined completely.
- On the contrary, $H_l \neq \varnothing$ meaning that the S_l set is not refined properly and yields improper results.

Runtime of KNN algorithm: This algorithm has the capability to yield more than half of the partitions in less time. It's been proved in several knowledge graphs that contain millions of triples and yields results within a few seconds or minutes. The core factor behind the efficiency of the algorithm is because of the usage of lazy joins for delivering results [Fer18].

Task of Link Prediction in KNN: The core concept behind the link prediction is to predict the missing node in a triple of the form (e_i, r_k, e_j). We can predict either the head e_i or the tail e_j of the triple. In this paper, we concentrate on predicting the tail e_j node. Author's approach of link prediction is motivated by the work of Denoeux [Den08]. Denoeux explains a k-NN clustering rule based on Dempster-Shafer theory.

Steps to be followed in Link Prediction:

- Each k nearest neighbour x_l of x which is to be partitioned is used as a support that x belongs to class c_l of x_l. D-S theory helps us to combine all the k pieces of support into a global support which defines a measure of *belief* of each class.

- For each tail node e_j, we rank each of the k nearest neighbour to the node e_j based on the degree of belief. We finally sort it in the decreasing order of beliefs. The equation to calculate belief of each e_j is given by
  $Bel_j = 1 - \prod_{l \in 1 \ldots L} (1 - \alpha_0 \phi_{l,j} e^{-} d_l)$
  where constant $alpha_0$ determines the maximum degree of belief. Lower than 1 means uncertainity about the inferred triple (e.g. 0.80). The degree of belief plummets exponentially with the distances between the nodes.

Performance evaluation parameters:

1. **HITS@N:** Defined as a proportion of inferring the correct tail node which appear in the first N entities. The value lies between 0 and 1.
2. **Mean Reciprocal Rank (MRR):** Defined as the average of the inverse of the rank of the correct entity. The value lies between 0 and 1.

Dataset	Entities	Relations	Train edges	Valid. edges	Test edges
FB15k-237	15,541	237	272,115	17,535	20,466
JF17k	28,645	322	171,559	-	66,615
Mondial	2,473	20	7,979	778	970

Figure 3.3: statistics of datasets

Consider fig 3.3 [Fer19], it provides the facts regarding entities, relations, trained and valid edges along with test edges of various datasets. The datasets are as follows:

1. **FB15k-237:** It is a more refined version of FB15k by removal of inverse triples i.e., trivial inferences. Triples are taken from Freebase KG. FB15k-237 dataset is evaluated for link prediction [BUGD+13].
2. **JF17k:** It consists of binary as well as N-array relations. However, we concentrate on only binary relations for link prediction [WLM+16]. We look at the performance evaluation of KNN on several datasets in the results section.
3. **Mondial:** It is a subset taken from Mondial database which contains geographic related data [May99]. We remove the triples with N-array relations and the triples whose edges contain numbers and dates for link prediction.

4 Results

Approach	MRR	Hits@1	Hits@3	Hits@10
Freq	.236	.175	.253	.356
AMIE+	.143	.096	.155	.241
(from [14])	-	.174	-	.409
DistMult*	.191	.106	.207	.376
ComplEx*	.201	.112	.213	.388
HolE*	.222	.133	.253	.391
TransE*	.233	.147	.263	.398
R-GCN*	.248	.153	.258	.414
ConvE**	.325	.237	.356	.501
CNN 0.01s (ours)	.250	.186	.268	.377
CNN 0.1s (ours)	.264	.198	.284	.395
CNN 1s (ours)	.286	.215	.311	.428

Figure 4.1: Results on FB15k-237 for CNN approach with timeouts(0.01s, 0.1s, 1s)

Consider fig 3.3, the aim of the test is to infer the missing node i.e., *missing entity* either a tail or an head from a relation of the triple. The test entity is also called as a known node. Evaluation is performed using 4 performance parameters i.e., MRR and Hits@1,3,10 described in fig 4.1 [SKB+18].

Test method:

- KNN approach does not require training i.e., an instance based approach. This approach has two parameters whose value is set before the execution such as the *depth* which provides details about the test node and the *timeout* which provides the allocated execution time.

- We set the value of $\alpha_0 = 0.95$ i.e., the degree of belief and use all the evaluated nearest neighbours for the purpose of ranking.

- The experiments were executed on Fedora 25, with CPU Intel(R) Core(TM) i7-6600U @ 2.60 GHz, and 16 GB DDR4 RAM. This approach worked efficiently as the main memory conceded under 1.5% i,e,. about 240 Mb [Fer19].

Result: From the fig 4.1, *Freq* also known as baseline Freq is used to rank nodes based on global statistics and is independent of test entity. Other approaches except AMIE+ uses latent based approaches and the latter uses rule-based approach. Based on the statistics, we notice latent based approaches perform well compared to rule-based approaches. CNN surpasses all the state-of-the-art approaches except ConvE and it's MRR lies between R-GCN (-3.8%) and Conv (+3.8 %). CNN surpass *Freq* at all timeouts and learns beyond the global facts.

Relation	#heads	#tails	MRR_{Freq}	MRR	Hits@1	Hits@3	Hits@10
profession	4245	150	*.434*	.601	.455	.694	.874
gender	4094	2	*.882*	.899	.798	1	1
nationality	4068	100	*.720*	.772	.662	.866	.941
award	3386	406	*.080*	.270	.154	.296	.511
type_of_union	3033	4	*.971*	.971	.942	1	1
place_of_birth	2613	704	*.155*	.183	.100	.235	.359
place_lived	2519	804	*.172*	.194	.108	.239	.344
film/genre	1875	123	*.315*	.380	.226	.429	.711
film/language	1735	59	*.744*	.759	.688	.790	.911
film/country	1708	61	*.685*	.701	.573	.809	.931

Figure 4.2: Results of tail prediction on some of the frequent queries in FB15k-237 using CNN approach [Fer19]

Timeout	#concepts	max. belief	MRR
0.01	11.8	.467	.235
0.1	49.1	.795	.264
1	219.6	.943	.286

Figure 4.3: tail prediction at depth = 1 on FB15k-237 using CNN approach [Fer19]

Deeper Analysis: Consider the fig 4.2, we notice predicting tail is effortless compared to predicting head because the relations follows a deterministic pattern. We have also compared it with MRR of Freq-baseline. CNN outperforms Freq-baseline in all of it's relations. The factors influencing prediction are as follows:

1. **Timeout:** Table in fig 4.2 portrays that with 1% of the largest timeout, the MRR already reach closer to largest MRR i.e., 82% although only 5% of facts have been computed. We are able to predict the nodes with minimum effort i.e., early approximations.

2. **Depth:** Table in fig 4.3 portrays the influence of prediction with respect to depth of the nodes along with it's edges. FB15k-237 has roughly 750 edges at depth 1, 20,000 edges at depth 2, and so on. More time is required to compute triples at greater depths. Greater depth provides more information and paves way to discriminate each triple. At fewer depth, CNN approach is able to predict enough triples although at greater depths, the scope of improvement i.e., standard deviation is just 0.005.

5 Conclusion and Future work

Author uses one of the symbolic approaches to tackle the problem of link prediction. KNN approach is simple but effective algorithm mainly because there is no training phase, which is quiet time intensive. Evaluations on FB15k-237 dataset yielded excellent results compared to other state-of-the-art latent based and rule based tools. The present work concentrates on binary-relationships in KG and the future work will be focused on dealing with N-array relations. Relatively low priority focus would be optimizing the partition algorithm and the usage of parallel systems for computation.

Bibliography

[BKS13] Joanna Biega, Erdal Kuzey, and Fabian M Suchanek. Inside yago2s: a transparent information extraction architecture. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 325–328, 2013.

[BUGD+13] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, pages 2787–2795, 2013.

[Den08] Thierry Denoeux. A k-nearest neighbor classification rule based on dempster-shafer theory. In *Classic works of the Dempster-Shafer theory of belief functions*, pages 737–760. Springer, 2008.

[Fer18] Sébastien Ferré. Answers partitioning and lazy joins for efficient query relaxation and application to similarity search. In *European Semantic Web Conference*, pages 209–224. Springer, 2018.

[Fer19] Sébastien Ferré. Link prediction in knowledge graphs with concepts of nearest neighbours. In *European Semantic Web Conference*, pages 84–100. Springer, 2019.

[HFD12] Alice Hermann, Sébastien Ferré, and Mireille Ducassé. An interactive guidance process supporting consistent updates of rdfs graphs. In *International Conference on Knowledge Engineering and Knowledge Management*, pages 185–199. Springer, 2012.

[LNK07] David Liben-Nowell and Jon Kleinberg. The link-prediction problem for social networks. *Journal of the American society for information science and technology*, 58(7):1019–1031, 2007.

[May99] Wolfgang May. Information extraction and integration with florid: The mondial case study. Technical report, Technical Report 131, Universität Freiburg, Institut für Informatik, 1999.

[NMTG15] 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, 2015.

[SKB+18] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, pages 593–607. Springer, 2018.

[WLM+16] Jianfeng Wen, Jianxin Li, Yongyi Mao, Shini Chen, and Richong Zhang. On the representation and embedding of knowledge bases beyond binary relations. *arXiv preprint arXiv:1604.08642*, 2016.