Structural Characteristics of Knowledge Graphs Determine the Quality of Knowledge Graph Embeddings Across Model and Hyperparameter Choices

Jeffrey Sardina[1], Declan O'Sullivan[1]

[1] *Trinity College Dublin, College Green, Dublin, Address, Ireland*

Abstract

The realm of biomedicine is producing information at a rate far beyond the capacity of clinicians, researchers, and machine learning experts to analyse in full. Recently, developments in Knowledge Graphs (KGs) have facilitated the representation of all this information in an easily-integrable and easily-queryable format. With increasing academic and clinical interest in Knowledge Graph Embeddings (KGEs), various KGE models have been developed to allow machine learning to efficiently run on these large Knowledge Graphs and predict new, previously unseen information about the domain. However, the need to validate hyperparameters for every new dataset, especially considering the time and expertise needed for validation and model training, have limited the use of KGEs in bi-ology to those who have expertise in machine learning and knowledge engineering. This research presents a framework by which the effect of hyperparameters on model performance for a given KG can be modelled as a function of KG structure. The presented evaluation of the framework finds a clear effect of graph structure on hyperparameter fitness. This leads to the conclusion that more re-search into cross-dataset hyperparameter prediction and re-use holds promise for increasing the accessibility and usability of KGEs for biomedical applications.

Keywords

Knowledge Graphs, Hyperparameters, Knowledge Graph Embeddings

1. Introduction

Cancer biology and biomedical sciences are being revolutionised by Big Data. From projects such as Bio2RDF [1], the International Cancer Genome Consortium [2], the 1,000 Genomes Project [3], and TumorMap [4], Big Data has become a centrepiece of biomedical research. With the ever-increasing magnitude of these datasets, several approaches have been taken to analyse and utilise them in full. Some projects, such as TumorMap, have focused on transforming the available data into a simpler for-mat through dimensional reduction mechanisms, accepting a degree of information loss in exchange for easier usability [4]. On the other hand, recent Linked Open Data (LOD) systems have attempted to represent the entirety of the data in an easily-queryable graph-based format [1, 5–7]. Among the projects that have taken this approach is Bio2RDF, a LOD data store that incorporates data from many different biological and biomedical datasets into a graph format [1]. Other groups have followed up upon such projects with graph-based machine learning methods called Knowledge Graph Embeddings (KGEs) to process entire LOD datasets at once [6, 8, 9].

However, using Knowledge Graph Embeddings (KGEs) on Knowledge Graphs (KGs) requires the selection of hyperparameters to the models, and proper selection of hyperparameters is critical to enabling the model to best learn the data at hand [10, 11].

This research addresses the question of whether the hyperparameters to KGE models on biological LOD proceed from the structure of the data. The data suggests that graph structure indeed is the

SeWebMeDa, May 29–29, 2022, Hersonissos, Greece
EMAIL: sardinaj@tcd.ie
ORCID: 0000-0003-0654-2938 (A. 1); 0000-0003-1090-3548 (A. 2)

driving force behind hyperparameter preference. Our characterisation of the interaction between graph structure and model performance in the context of an arbitrary hyperparameter sets suggests that model performance for given hyperparameter sets can be predicted based on graph structure, in the absence of data on the hyperparameters themselves.

It is anticipated that relating hyperparameters and KGE performance to structure would allow predicting and explaining why certain models perform better than others and would allow these results to be generalised to similarly structured graphs from diverse domains. While in this work the cancer biology and biomedical fields are used—due to their importance and due to the lead author's previous experience in both—the goal of explaining results in terms of KG structure allows for wider generalisation.

The characterisation achieved in this research, while not definitive, calls for further research into the possibility of allowing relational learning algorithms to operate on similarly-structured datasets using a consistent set of known hyperparameters, and on whether predicting these hyperparameters from graph structure along without a hyperparameter search may be possible. Such a system would allow biological LOD datasets to be analysed much more quickly, without the need for running a full search for optimal hyperparameters on every new dataset.

Please note that in this paper, references to "hyperparameters" refer not to only the parameters given to a KGE / relational learning model (such as the learning rate), but also to the choice of the models themselves; while differs from the formal definition of hyperparameters, it results in a simpler phrasing of the model choices made.

The remained of this paper is structured as follows: Section 2 gives a background on KGs, KGEs, and the gap in the current state-of-the-art that this research seeks to fill. Section 3 details the methods by which experiments were performed, and Sections 4 provides the results of these experiments. Section 5 concludes the paper and gives a discussion of the major findings, as well as the limitations of this study.

2. Background

2.1. Knowledge Graphs

The most popular KG format is RDF, the Resource Description Framework [12]. Like most KG formats, the smallest unit of information in the graph, a set of two entities and one predicate (or relationship) that links them [12]. As thus, Every KG is a special case of a graph G(V, E) with vertices V and edges E, where every vertex and edge is involved in at least one triple. Within the context of any one triple, the first entity is called the subject or the head, and the second entity the object or the tail [12]. This can be abbreviated simply as *(subject, predicate, object)*.

In this configuration, RDF triples mirror simple linguistic statements. For example, the statement "P53 is a protein" could be modelled as the triple *(p53, is-a, protein).*

In the RDF format, subject, predicates, and objects are often represented by URLs which allows for entities and predicates to be easily reused and dereferenced, either within one data source or between different data sources [12]. The ability to reuse entities and relationships means that various triples can be linked and connected logically to each other, either by sharing a head, a tail, or both. This allows many large KGs – to connect data from multiple datasets with relative ease in RDF.

2.2. Knowledge Graph Embeddings

2.2.1. Representing Vertices and Edges

Repeating entities as both a subject and an object allows linking triples to each other. While this can be used in many cases for logical reasoning using formal rules to ex-tract more information from the graph [12], this property also allows machine learning techniques to operate on the graph and learn to distinguish and related entities based on the triples they are involved in [13, 14]. These

machine learning techniques are referred to as relational machine learning since they learn based on these relations [13].

The output of this relational learning is a set of Knowledge Graph Embeddings, where entities are typically represented as vectors in Rn and relationships are represented by transformations on them [13, 14]. These relationships are structured (for example) as functions that map Rn onto Rn, allowing them to convert from subjects to the expected objects of that relationship (the so-called "link prediction task") [13, 14].

The choice of the dimension into which the entities are placed, as well as the choice of what sort of transformation is used to model the relationship (such as vector addition or matrix multiplication) are model design choices that must be investigated by developers to find the optimal combination for a given data set [13, 14].

2.2.2. Training on Negative Samples

In order for the KGE model to effectively learn to predict true triples and reject false ones, it must be trained not only on the known-true triples but also on known-false triples [13, 14]. This is done using a technique called negative sampling [13, 14].

There are various assumptions about the data that researchers can make to produce negative samples; the most common is Local Closed-World Assumption, which claims that if a given subject and predicate are observed with a certain set of objects, then that subject and predicate only ever match to objects of those types [13, 14]. This is a specific case of the Open-World Assumption, also commonly used, which assumes that there may be arbitrarily many true statements which are not contained in the KG [13, 14]. However, the Open-World Assumption gives no way of predicting which triples that are not in the KG are true and which are false, and is less commonly used [13, 14].

Choosing by what method or methods to sample negatives, and how many to sample, are also hyperparameters to KGE models.

2.2.3. The TransE Model for KGEs

The most basic conceptualisation of KGEs is the TransE model [13, 14]. Under the TransE model, nodes are embedded as vectors, and relationships as vector displacements between those nodes. If si is the embedding of a subject node, pi the embed-ding of a relationship, and oi the embedding of an object node, then TransE attempts to enforce the following equality as closely as possible:

$$s_i + p_i = o_i \quad (1)$$

In essence, this gives a very intuitive definition of embeddings: the subject (entity) plus the predicate (how the subject relates to the predicate) should be close or equal to the object [13, 14].

A simple example of embedding a KGE into 2-space is given in Figure 1.

simple example of embedding a KGE into 2-space is given in Figure 1.

Original Knowledge Graph **Embedded Knowledge Graph**

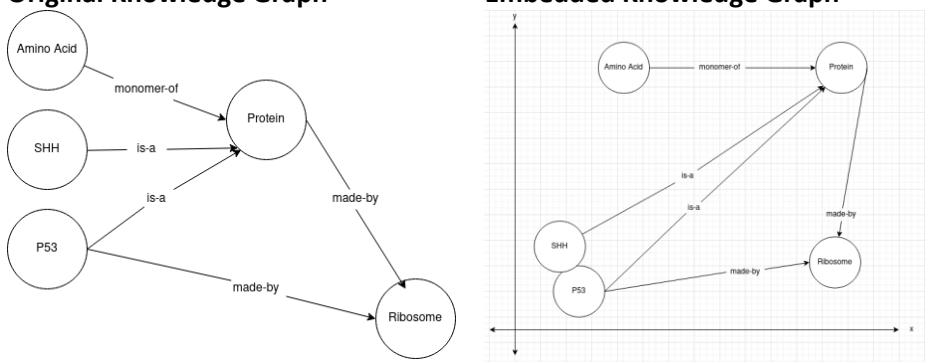

Fig. 1. Visualisation of KGEs (TransE)

Link prediction in TransE is formulated as taking the translation of some subject *s* under some predicate *p*, resulting in a vertex *v'*. The closest other vertex *o* to *v'* in R^n is predicted to be the object of the relationship *(s, p, o)*. A visualisation of this method is given in Figure 2.

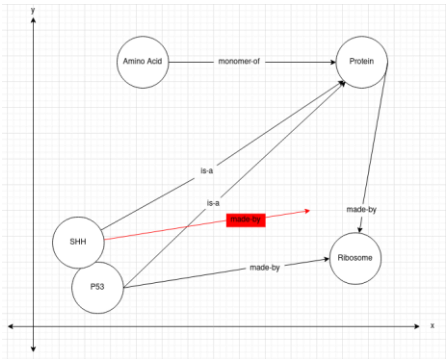

Fig. 2. A visualisation of link prediction in TransE, where the object Ribosome would be predicted to produce the triple *(SHH, made-by, Ribosome)*. The relationship made-by is shown as a red arrow transforming the subject SHH to a region closest to the predicted object, Ribosome.

2.2.4. Other KGE Models

Many other KGE models exist; however, they all follow the same basic idea of using some transformation (represented by an edge) on subjects to produce embeddings that can be matched by some metric to the correct objects [13, 14].

In terms of model definition, it is very common that the operator used to represent edges, the comparison metric (i.e., Euclidean distance or cosine similarity) between transformed subjects and objects, the loss function used to score embeddings, and regularisation terms added differ between various models [10, 13, 14]. However, a large variety of even more diverse KGE methods exist [13, 14].

2.3. Gap in the State-of-the-Art

The KGE models that have been applied have focused on producing a set of embeddings either for use as feature-vector inputs to a different machine learning application [15, 16] or directly for novel link prediction within the learned dataset [16]. Both run into the inevitable issue of selecting hyperparameters, which often involves a time-consuming brute-force search [16].

The recent trend towards both more advanced visualization for KGs in bioinformatics, as in the transition from the LTCGA to BIOOPENER [5–7], as well as interest in automated KGE models [15, 16] are producing results that commonly match or exceed other modern approaches in data visualization and prediction [6, 16]. In the case of this research, focus was given specifically to how to reduce the up-front barrier to entry posed by hyperparameter selection.

While various articles have established that KGEs are very sensitive to good hyperparameter choice [16], characterized the most important meta-elements of graphs such as structure and provenance [17], and noted the effect of graph structure on embeddings [18], no attempts have been made to relate these hyperparameters to KG structural features. Doing so would thus be a contribution both to the field of knowledge-graph machine learning and to the of KGE-based bioinformatics.

3. Background

3.1. Gap in the State-of-the-Art

Selection of data sources occurred in two steps: selection of a multi-dataset LOD mashup, and selecting datasets from within that mashup. Selection of data from a single mashup rather than from solitary KGs was done for four reasons: simplicity, ease of reproducibility, relevance, and consistency.

Simplicity and ease of reproducibility for LOD-based projects go hand in hand. Mashup systems such as Bio2RDF are intended to be used in many different contexts and by different applications [1]. Moreover, they are designed for easy access of their components by researchers [1]. Both attributes make the data attractive in terms of simplicity: all the data is easily obtained from a single place. Moreover, ease of access to the data is a basis for ease of reproducibility, since other groups who wish to reproduce the results of this work need only reference data from a single location rather than many.

In addition, these larger data mashups tend to have much higher overall relevance. Bio2RDF, for example, was constructed from the most common biological datasets [1], and now contains a total of 35 biological datasets [19].

From within Bio2RDF, a small selection of datasets was chosen based on their relevance to biomedical research, as well as on their overall size.

In terms of the first criterion, datasets from Bio2RDF that were immediately relevant to cancer and biomedical research were selected given the lead author's experience and interest in these domains. These domains, of course, are very broad and can encompass a variety of types and sources of data. Specifically, in order to select the datasets most relevant to these categories, datasets containing information on drugs, molecular biology, clinical data, and genetics were selected as the most highly relevant. Datasets containing data exclusively from non-human animals were excluded to focus more clearly on biomedical modelling in the human context. This left a list of 15 potential datasets.

Once this list of datasets was obtained, a subset of datasets to be used in analysis was selected based on dataset size. This filter was introduced for purely practical reasons: larger datasets take significantly longer to pre-process and train, even at low epochs.

In order to strike a balance between including enough datasets in the pool for analysis and minimizing the overall computational time and power spent, any datasets in excess of 20GB were removed from consideration. Moreover, datasets measuring under 1 MB were removed for containing too little information, since the goal of this work is to focus on big data rather than learning from small KGs. This resulted in a list of 9 datasets from the Bio2RDF mashup being used: BioPortal (18.3 GB), Database of single nucleotide polymorphism (DBSNP, 2.9 GB), DrugBank (1.6 GB), Gene Ontology Annotation (GOA, 17.1 GB), HUGO Gene Nomenclature Committee (HGNC, 844.8 MB), the Kyoto Encyclopedia of Genes and Genomes (KEGG, 18.1 GB), the Life Science Resource Registry (LSR, 12.1 MB), Online Mendelian Inheritance in Man (OMIM, 2.2 GB), and Pharmacogenomics Knowledge Base (PharmGKB, 1.5 GB).

3.2. KGE Implementation

In this research, KGE models were implemented in PyTorch-BigGraph (PBG) [10]. PBG allows the user to define a KGE model by choosing an operator (to transform subject embeddings to object embeddings), a comparator (to define how to measure closeness of two embeddings) and a loss function (to optimize the model) [10].

Importantly, PBG also allows the main graph to be split into partitions. Each partition is loaded into memory one at a time, and each one represents the fraction of the graph that can fit in system memory at once [10]. PBG then handles communicating the results of training on different partitions between partitions and uses this to create an overall KGE model [10]. As a result, PBG is well-suited for large-scale biomedical (or other) LOD datasets; an example of its use may be seen in [20].

The authors showed that, for large KGs, this system approximates the results that would be obtained using an identical configuration on non-partitioned graphs [10]. However, they did note that creating partitions on smaller graphs could have some negative effects on embedding quality [10].

3.3. Hyperparameter Selection

In order to select hyperparameters, a modified version of the grid search was used. In a traditional grid search, all values of all hyperparameters in question are varied over a grid, and the best choice from among them is chosen. However, the large number of KGE models and hyperparameters involved made such an approach infeasible (for a listing of hyperparameters involved, see Table 2). Thus, an arbitrary set hyperparameters values given in [21] were used to initialize the model.

Only five of the 9 total datasets were used in the initial round of hyperparameter selection; these were BioPortal, DBSNP, DrugBank, OMIM, and PharmGKB. This approach was undertaken so that the resulting hyperparameter configurations could be run on datasets for which they had not been created as a measure of how well the hyperparameter configurations worked across different datasets.

Three grid searches were then carried out: in the first, model-related hyperparameters were varied. These were, specifically: comparator, learning rate, loss function, operator, and regularisation coefficient. It should be noted that, due to the design of PBG, a regularization coefficient hyperparameter is not given to KGE models using the affine operator [21]. As such, this combination of operator and regularization coefficient was not allowed when searching for optimal hyper parameter calculations.

In the second round, hyperparameters relating to batching were varied; these were batch size, the number of batch negatives to use, and the number of uniformly sampled negatives to use. Finally, in the third round, the number of epochs and embedding dimensions were varied.

Table 1
A summary of the hyperparameter search rounds.

Gird Search Round	Hyperparameters Involved
1 (Model-related hyperparameters)	comparator, learning rate, loss function, operator, regularisation coefficient
2 (Batch-related Hyperparameters)	Batch size, number of batch negatives, number of uniform negatives
3 (Epochs and Dimensions)	Embedding dimension, number of epochs

In addition, rather than conducting the hyperparameter search on the entire dataset, datasets were subsetted randomly in order to make the search feasible in the available time. In order to do this, the decision of how large to make the subsets was critical to ensuring that they could well represent the data from which they were drawn. All subsets taken were taken in a single-pass traversal of the graph in which a triple was randomly chosen with probability equal to the desired number of triples divided by the total number of triples in the graph. The desired number of triples was set to 4,000.

For all rounds, model performance was measured using the "r1" metric, which is the probability that a true triple would be preferred over all of the negative triples created samples for it under the Local Closed-World Assumption during link prediction [21].

4. Background

4.1. Hyperparameter Sets Selected

Ultimately, two different sets of hyperparameters were created: one for BioPortal, and one for DBSNP, DrugBank, OMIM, and PharmGKB. The reason for these two sets was that, in each of the hyperparameter validation rounds, in almost all cases the distribution of r1 scores given different hyperparameter combinations for DBSNP, DrugBank, OMIM, and PharmGKB matched very closely, while BioPortal did not follow this trend. The hyperparameters selected are shown in Table 2.

Table 2
Hyperparameter configurations selected during the modified grid search.

BioPortal ("BioPortal Configuration")		**DBSNP, DrugBank, OMIM, and PharmGKB ("General Configuration")**	
Hyperparameter	**Value**	**Hyperparameter**	**Value**
Model-related Hyperparameters		***Model-related Hyperparameters***	
Comparator	L_2	Comparator	L_2
Learning rate	1e-2	Learning rate	1e-4
Loss function	Ranking	Loss function	SoftMax
Operator	Translation	Operator	Translation
regularisation coefficient	1e-1	regularisation coefficient	1e-3
Batch-related Hyperparameters		***Batch-related Hyperparameters***	
Batch size	1000	Batch size	500
Number of batch negatives	500	Number of batch negatives	100
Number of uniform negatives	100	Number of uniform negatives	250
Epochs and Dimensions		***Epochs and Dimensions***	
Embedding dimension	200	Embedding dimension	100
Number of epochs	50	Number of epochs	200

Once all the hyperparameters had been obtained, all datasets were run using both sets of hyperparameters. The output r1 scores for all runs are shown in Table 3.

Table 3
R1 Scores of all datasets using both hyperparameter configurations.

Dataset	**r1**	**Dataset**	**r1**
Under BioPortal Configuration		*Under General Configuration*	
BioPortal	0.3894	BioPortal	0.3912
DBSNP	0.1907	DBSNP	0.1306
DrugBank	0.2068	DrugBank	0.1067
OMIM	0.1381	OMIM	0.1209
PharmGKB	0.1312	PharmGKB	0.0757
GOA	0.3099	GOA	0.2373
HGNC	0.1257	HGNC	0.0711
KEGG	0.2948	KEGG	0.0938
LSR	0.1873	LSR	0.1460

Interestingly, the general configuration yielded its best scores on datasets it had not been created to accommodate (BioPortal and KEGG). Those datasets it was trained on were (DBSNP, DrugBank, OMIM, and PharmGKB) universally had lower performance when trained on that hyperparameter set.

The BioPortal configuration outperformed the general configuration on all datasets except one: BioPortal. While this difference was small and quite possibly insignificant, its variance was not estimated. In any case however, the BioPortal configuration was not clearly better for BioPortal itself.

The departure of these results from the expected ones—that the BioPortal configuration would be optimal for BioPortal by a large margin and the general configuration would be similarly superior for DBSNP, DrugBank, OMIM, and PharmGKB from which it was created—suggest that the selected hyperparameter values are not optimal.

However, it was noted that even in the absence of optimal or near-optimal hyperparameters, the data can be interpreted as coming from arbitrary hyperparameter selections, making no assumptions about the goodness (or lack thereof) of the model choices, it is under this assumption that the remaining analysis was carried out.

4.2. Relating KG Structure, Hyperparameters, and Model Performance

Relating KG structure to model performance and hyperparameters was formulated as a regression problem: given a set of structural characteristics, predict the r1 scores of a model under a single hyperparameter set. Each prediction was made in the context of data from all datasets under a single hyperparameter configuration only.

The first step in this process was to identify relevant structural features from each KG. Since it has been noted that KG connectivity—particularly centrality—impacts KGEs [18], two different methods to examining centrality were applied The first was by measuring the counts and proportions of sources (nodes that are only ever subjects), sinks (nodes that are only ever objects) and repeats (nodes that and a subject and an object at least once in the KG). The second was by measuring the distribution of the centrality of nodes, where centrality was calculated as the total degree of a node. The outputs of both of these methods are shown in Tables 4 and 5.

Table 4
A summary of KG structure on basis of the sources, sinks, and repeats. These are given as raw counts and as ratios to the number of triples in the KG.

Dataset	Number of Sinks	Number of Sources	Number of Repeats	Ratio of Sinks to Triples	Ratio of Sources to Triples	Ratio of Repeats to Triples
Used in the searches						
BioPortal	13623972	4385915	4577592	0.1496	0.0482	0.0503
DBSNP	2981629	586130	550671	0.2360	0.0464	0.0436
DrugBank	2079154	414454	399251	0.3200	0.0638	0.0614
OMIM	4507308	1121071	1122401	0.4291	0.1067	0.1068
PharmGKB	2811465	641974	521171	0.3952	0.0902	0.0733
Not used in the searches						
GOA	15674396	3073804	3091543	0.1814	0.0356	0.0358
HGNC	1897778	415958	416245	0.4447	0.0975	0.0975
KEGG	27216889	8622153	8504190	0.3051	0.0967	0.0953
LSR	26089	6137	6130	0.4298	0.1011	0.1010

Table 5

Centrality statistics for all datasets, where centrality is measured by node degree. Average is the average centrality, Q1 the 1st quartile of centrality, Median its median, Q3 the 3rd quartile, Max the single highest centrality seen, and Max to triples ration the ration of the highest centrality seen to the number of triples in the KG

Dataset	Average	Q1	Median	Q3	90%	Max	Max to Triples Ratio
Used in the searches							
BioPortal	9.2939	1	2	6	15	6504014	0.0714
DBSNP	7.0801	1	1	1	17	697049	0.0552
DrugBank	5.2104	1	1	1	12	428085	0.0659
OMIM	3.7322	1	1	1	7	1185612	0.1129
PharmGKB	4.1191	1	1	1	8	666924	0.0938
Not used in the searches							
GOA	9.2163	1	1	1	16	6080694	0.0704
HGNC	3.6888	1	1	1	8	416264	0.0975
KEGG	4.9774	1	2	2	12	12115498	0.1358
LSR	3.7672	1	1	2	8	4359	0.0718

From these features, a subset was selected as inputs to a regressor. The structural features selected to measure the effect of the high prevalence of sinks versus sources and repeats were the ratios of sinks and repeats to triples. The ratio of sources to triples was not included, since in all cases it was nearly identical to the ratio of repeats to triples and thus was redundant. Moreover, adding in more features on datasets with few data points can lead to machine learning models overfitting by memorizing data rather than learning general trends, which would make interpretation of the results less clear.

The ratio of the maximum centrality to the number of triples, as well as median, 3rd quartile, and 90th percentile centralities were used to represent the effects of centrality. Notably, the first quartile was not included since it was identical in all datasets. Other values were not included because they varied similarly to data already in the dataset and would result in introducing too many features and potentially overfitting the data.

Data from all datasets was normalised prior to being input into the regression models, and all regression models were run with 5-fold cross-validation using an L1 (or “Lasso”) penalty to select the regularisation coefficient. The Lasso regression penalty was chosen since it tends to drive the values of parameters that are not needed in the regression decision to zero, thus providing an easy tool for the detection of which structural elements are important and which are of no use for gaining predictive power. R^2 scores were obtained by from the final Lasso model on the training data.

Since all input values were normalized, the parameter values themselves can be used as an estimate of their importance to the regression decision within the context of a single model: those with higher values represent structural features that have a created correlation to the final r1 score obtained by the KGE model. It must be noted, however, that care must still be taken in this interpretation of the parameter values, especially since interaction terms were not considered and thus some effects of the variance of each parameter on the model may not be fully contained in the given data.

The results of Lasso regression and the final Lasso models are given in Tables 6 – 8.

Table 6

R2 values of a Lasso regressor trained the given type of structural data to predict r1 scores as obtained for each dataset by the specified hyperparameter configuration.

Data used	BioPortal Configuration	General Configuration
Sink-repeat ratios	0.9076	0.6278
Centrality distribution	0.8282	0.8129

Table 7
The coefficients in the Lasso model to each parameter from the source-sink structural statistics data.

	Sinks:Triples ratio	Repeats:Triples ratio	Median centrality
BioPortal configuration	-0.1359	0.0674	0.0290
General configuration	-0.1089	0.0375	0

Table 8
The coefficients in the Lasso model to each parameter from the centrality statistics data.

	Median centrality	3rd quartile centrality	90th percentile centrality	Max centrality to Triples Ratio
BioPortal configuration	0.0290	0.0237	0.0373	0
General configuration	0	0.0675	0.0199	-0.0115

All models examined achieved R2 values of at least 0.60; all but one was above 0.80. While all the models based on sink and repeat frequencies used both features in the final regression model, none of the models using centrality distribution statistics used all the available features; in both cases on of then was ignored by the regressor.

These results suggest that, for datasets trained on the same hyperparameters, their structures correlate very well with how well they perform under that hyperparameter set. This suggests a possible effect of KG structure on how well a model performs given an arbitrary set of hyperparameters. Or, put differently, it suggests that the fitness of a hyperparameter set for a given KG, and ultimate KGE performance under that set, can be determined from the structure of the KG alone.

5. Discussion and Conclusions

5.1. Key Contributions

It is expected that this work contributes to relational learning and bioinformatics in two key ways.

First, it suggests that KGE performance is very responsive to structure, particularly with respect to proportions of sources, sinks, and repeated entities in the graph and to the distribution of the centrality of nodes in the graph.

It also demonstrates that the performance of a KGE with a given set of hyperparameters can be predicted with high accuracy considering only KG structure. This suggests that it would be possible to rapidly predict what

KGs may fit a given set of hyperparameters using only a linear regression model, rather than a time-intensive grid search.

The work also indicates two important future directions, which if followed could provide critical insights to the field. One on side, this research suggests that it may be possible to create a regression model that predicts model performance not only based off the KG structure, but also upon hyperparameters, in the absence of training. If done, this would allow for the rapid detection of optimal hyperparameter configurations—even those never examined before—without the need to do a grid search. It is hypothesised that this model could then be applied to very rapidly find the ideal hyperparameter configuration for a KG with a given structure.

Secondly, it suggests that hyperparameter selection may be able to be formulated as a classification problem, mapping from KG structural statistics to one of several models and sets of hyperparameters, without any need for a gird search or traditional hyperparameter selection methods.

5.2. Limitations of this Work

Since the method for selecting hyperparameters was found to be sub-optimal for the intended datasets, the hyperparameter sets produced are known to be imperfect. This is possibly a result of having subsetted the graphs, producing subgraphs that could be easily grid-searched, but whose structure was notably different in some respects from the structure of the original KG. Unfortunately, the extent by which they vary from the optimum configurations was impossible to estimate, as the optimal configurations are not known. As a result, the results of this study are interpreted as presenting data in the context of arbitrary hyperparameter configurations, rather than optimal or near optimal ones. However, the structural analysis of KGE scores under these configurations remains valid, because that analysis made no assumption of optimality of the configurations which it was predicting.

In addition, in this work only biomedical datasets were considered, and all these datasets were observed to have an extremely strong skewness of centrality values. Given the findings of this research, examining how other datasets with very different centrality distributions than those seen here interact with optimal hyperparameters to KGE models is expected to be of benefit to the relational learning and KGE fields.

Finally, in this work even the final hyperparameter configurations identified yielded low r1 scores which never exceeded 0.4. Scores observed in the hyperparameter search were similarly low, often significantly lower. However, the reason for these low scores was not directly identified. As outlined in Wang et al., KGs learn by understanding the relationships between entities [14]. This suggests, then, that entities with very few observed relationships would be relatively hard to learn to embed properly. Therefore, it stands to reason that an effect of structure would be observed here as well, very likely in terms of the number of sinks and sources in the graph relative to the number of triples in total. This effect may be partly agnostic to the hyperparameters involved, although this determination could not be made with the data available. Further research in this direction would be merited.

5.3. Final Observations

It is hoped that this work contributes to the understanding of hyperparameter choices not only in the realm of KGEs, but in the context of machine learning models generally. The finding that the structural elements of KGs are very highly predictive of model performance under different hyperparameter configurations suggests that data structure and model choice may be best understood in the context of each other.

Creating machine learning models by which structural elements could lead to optimal hyperparameter prediction and predictions of model performance is well merited. Moreover, it would be equally merited to extend this work to other machine learning domains, to understand if all

hyperparameters and model performance—or only those for KGEs—can be modelled as a function of dataset structure.

The author of this work hypotheses that such dataset-structure based approaches would yield fruitful results, advancing understanding of machine learning models and facilitating optimal hyperparameter selection in machine learning domains outside of KGEs alone. Furthermore, it is hypothesised that structure-based optimal or near-optimal hyperparameter determination may be possible even in the absence of any form of traditional hyperparameter search for KGEs, and it is suggested that further work in this area examine whether such approaches are effective and practical.

6. References

[1] 1. Belleau, F., Nolin, M.-A., Tourigny, N., Rigault, P., Morissette, J.: Bio2RDF: Towards a mashup to build bioinformatics knowledge systems. Journal of Biomedical Informatics. 41, 706–716 (2008). https://doi.org/10.1016/j.jbi.2008.03.004.

[2] 2. Zhang, J., Baran, J., Cros, A., Guberman, J.M., Haider, S., Hsu, J., Liang, Y., Rivkin, E., Wang, J., Whitty, B., Wong-Erasmus, M., Yao, L., Kasprzyk, A.: International Cancer Ge-nome Consortium Data Portal--a one-stop shop for cancer genomics data. Database. 2011, bar026–bar026 (2011). https://doi.org/10.1093/database/bar026.

[3] 3. The 1000 Genomes Project Consortium: An integrated map of genetic variation from 1,092 human genomes. Nature. 491, 56–65 (2012). https://doi.org/10.1038/nature11632.

[4] 4. Newton, Y., Novak, A.M., Swatloski, T., McColl, D.C., Chopra, S., Graim, K., Weinstein, A.S., Baertsch, R., Salama, S.R., Ellrott, K., Chopra, M., Goldstein, T.C., Haussler, D., Morozova, O., Stuart, J.M.: TumorMap: Exploring the Molecular Similarities of Cancer Samples in an Interactive Portal. Cancer Res. 77, e111–e114 (2017). https://doi.org/10.1158/0008-5472.CAN-17-0580.

[5] 5. Saleem, M., Padmanabhuni, S.S., Ngomo, A.-C.N., Almeida, J.S., Decker, S., Deus, H.F.: Linked cancer genome atlas database. In: Proceedings of the 9th International Conference on Semantic Systems - I-SEMANTICS ’13. p. 129. ACM Press, Graz, Austria (2013). https://doi.org/10.1145/2506182.2506200.

[6] 6. Jha, A., Khan, Y., Mehdi, M., Karim, M.R., Mehmood, Q., Zappa, A., Rebholz-Schuhmann, D., Sahay, R.: Towards precision medicine: discovering novel gynecological cancer biomarkers and pathways using linked data. J Biomed Semant. 8, 40 (2017). https://doi.org/10.1186/s13326-017-0146-9.

[7] 7. Saleem, M., Kamdar, M.R., Iqbal, A., Sampath, S., Deus, H.F., Ngonga Ngomo, A.-C.: Big linked cancer data: Integrating linked TCGA and PubMed. Journal of Web Semantics. 27–28, 34–41 (2014). https://doi.org/10.1016/j.websem.2014.07.004.

[8] 8. McCusker, J.P., Dumontier, M., Yan, R., He, S., Dordick, J.S., McGuinness, D.L.: Find-ing melanoma drugs through a probabilistic knowledge graph. PeerJ Computer Science. 3, e106 (2017). https://doi.org/10.7717/peerj-cs.106.

[9] 9. Hasan, S.M.S., Rivera, D., Wu, X.-C., Durbin, E.B., Christian, J.B., Tourassi, G.: Knowledge Graph-Enabled Cancer Data Analytics. IEEE J. Biomed. Health Inform. 24, 1952–1967 (2020). https://doi.org/10.1109/JBHI.2020.2990797.

[10] 10. Lerer, A., Wu, L., Shen, J., Lacroix, T., Wehrstedt, L., Bose, A., Peysakhovich, A.: PyTorch-BigGraph: A Large-scale Graph Embedding System. arXiv:1903.12287 [cs, stat]. (2019).

[11] 11. Ali, M., Hoyt, C.T., Domingo-Fernández, D., Lehmann, J., Jabeen, H.: BioKEEN: a library for learning and evaluating biological knowledge graph embeddings. Bioinformatics. 35, 3538–3540 (2019). https://doi.org/10.1093/bioinformatics/btz117.

[12] 12. Bizer, C., Heath, T., Berners-Lee, T.: Linked Data - The Story So Far: International Journal on Semantic Web and Information Systems. 5, 1–22 (2009). https://doi.org/10.4018/jswis.2009081901.

[13] 13. Nickel, M., Murphy, K., Tresp, V., Gabrilovich, E.: A Review of Relational Machine Learning for Knowledge Graphs. Proc. IEEE. 104, 11–33 (2016). https://doi.org/10.1109/JPROC.2015.2483592.
[14] 14. Wang, Q., Mao, Z., Wang, B., Guo, L.: Knowledge Graph Embedding: A Survey of Approaches and Applications. IEEE Trans. Knowl. Data Eng. 29, 2724–2743 (2017). https://doi.org/10.1109/TKDE.2017.2754499.
[15] 15. Mohamed, S.K., Nounu, A., Nováček, V.: Biological applications of knowledge graph embedding models. Briefings in Bioinformatics. 22, 1679–1693 (2021). https://doi.org/10.1093/bib/bbaa012.
[16] 16. Celebi, R., Uyar, H., Yasar, E., Gumus, O., Dikenelli, O., Dumontier, M.: Evaluation of knowledge graph embedding approaches for drug-drug interaction prediction in realistic set-tings. BMC Bioinformatics. 20, 726 (2019). https://doi.org/10.1186/s12859-019-3284-5.
[17] 17. Ben Ellefi, M., Bellahsene, Z., Breslin, J.G., Demidova, E., Dietze, S., Szymański, J., Todorov, K.: RDF dataset profiling – a survey of features, methods, vocabularies and applications. SW. 9, 677–705 (2018). https://doi.org/10.3233/SW-180294.
[18] 18. Sadeghi, A., Collarana, D., Graux, D., Lehmann, J.: Embedding Knowledge Graphs Attentive to Positional and Centrality Qualities. In: Oliver, N., Pérez-Cruz, F., Kramer, S., Read, J., and Lozano, J.A. (eds.) Machine Learning and Knowledge Discovery in Databases. Re-search Track. pp. 548–564. Springer International Publishing, Cham (2021). https://doi.org/10.1007/978-3-030-86520-7_34.
[19] 19. Bio2RDF Release 3, https://download.bio2rdf.org/files/release/3/release.html, last accessed 2022/02/24.
[20] 20. Fisher, J., Palfrey, D., Christodoulopoulos, C., Mittal, A.: Measuring Social Bias in Knowledge Graph Embeddings. arXiv:1912.02761 [cs]. (2020).
[21] 21. Welcome to PyTorch-BigGraph's documentation! — PyTorch-BigGraph 1.dev documentation, https://torchbiggraph.readthedocs.io/en/latest/index.html, last accessed 2022/02/24.