Evaluating a Proposed Link Interestingness Measure Using Oversampling of Triples in Knowledge Graphs

Tommy Lohn
Supervisor: dr. Michael Cochez
Second supervisor: Dimitrios Alivanistos

Vrije Universiteit Amsterdam, De Boelelaan 1105, 1081 HV Amsterdam, Netherlands
https://www.vu.nl/nl/index.aspx

Abstract. Knowledge Graph Embedding (KGE) models concern the representation of information from Knowledge Graphs (KGs) into vector representations. Although KGs are characterised by incompleteness, new knowledge is generated every day and thus, new facts can be added. However, experiments to obtain new knowledge are often costly and time consuming. Therefore, I propose a Link Interestingness measure that ranks potential triples based on how much information they provide. I evaluate the proposed measure based on the impact of adding triples as positive and negative triples to the training set, which will be referred to as positive and negative oversampling respectively. The results show that larger amounts of oversampling have an impact on the distribution of Link Prediction (LP) scores. This indicates that this proposed model should identify interesting links with some degree of significance. Further research could extent this model by looking at additional factors that impact link interestingness. In addition, future research could consider optimizing all parameters to increase the validity of the proposed model.

Keywords: Knowledge Graph · KG Embedding · Link Interestingness · Link Prediction

1 Introduction

The increasing amount of knowledge and information calls for means to accurately and efficiently model this knowledge. In recent research, attention has been drawn to representing knowledge in the form of KGs (Ji et al., 2021). KGs represent facts as triples where the subject and object represent entities and the predicate represents the relation between them. KGs can contain large amounts of facts about our world. As a result, KGs can be used for various applications in numerous domains (Kazemi & Poole, 2018). However, a major downside to the use of KGs is that KGs are incomplete, meaning that facts are missing from the KG.

One method to infer new links in KGs is LP. LP concerns the prediction of new links between entities based on existing links in the network (Y. Yang et al., 2015). LP largely serves two purposes. Firstly, LP can infer links that could be added in the future, therefore expanding the existing network. LP can also predict missing links. This second task aims at completing the existing network. Secondly, LP methods generate scores for the probability of a link being true. Although LP has valuable utilization, often not all links predicted are relevant (Pusala et al., 2017).

Link interestingness is a topic that is widely discussed in academic research (McGarry, 2005; Kontonasios et al., 2012; Pusala et al., 2017; Silberschatz & Tuzhilin, 1995). However, the definition of interestingness is still to be determined uniformly. In this paper, link interestingness is defined as the impact that a link has on the information gain within the KG. More specifically, this means that interesting links impart significant new knowledge to the KG. Although LP methods give predictions of the plausibility of a link, these predictions are not always reliable (Y. Yang et al., 2015). In real-world settings, especially biomedical settings, a reliable level of accuracy of these predictions is essential (Szilagyi et al., 2005). Experimental methods can be used to determine if a link is actually true or false. However, these experimental methods are often costly and time consuming.

Therefore, there is a need for an interestingness measure that can determine which links should be prioritised to experimentally test. In this work, these are thus links that increase the information gain significantly, when the plausibility of these links are known. This paper proposes an interestingness measure based on the difference in LP scores. Three situations are considered, namely scenario S,A and B, where in A the link is assumed to be true, in B the link is assumed to be false and in S it is unknown whether the link is true or false. This paper focuses on the following research question: to what extent can we evaluate the interestingness of a link by analyzing LP scores after oversampling potential triples?

The remainder of this paper is structured as follows. Section 2 gives an overview of background information and related works. Section 3 showcases the experimental setup and methodology. Section 4 displays an evaluation of the results. A discussion is provided in Section 5. Section 6 concerns the conclusion. Section 7 briefly highlights suggestions for future research.

2 Related Work

This section summarizes recent research that contributes in the understanding of this research experiment.

2.1 KGE models

KGE models are a form of machine learning tasks which learn to represent KGs as low-dimensional vectors or matrices. Q. Wang et al. (2017) provide a review of different knowledge embedding models, such as translational distance models and semantic matching models. KGE models represent KGs using entities and relations from the KG, which they then transform into vectors or matrices.

The authors mention several downstream tasks that could be performed using KGE models. Some of the most popular tasks include: LP, triple classification, entity classification, entity resolution and node classification.

In addition, the authors describe tasks that extent beyond the KGs. These tasks reach further than KGs and are used in a wide variety of domains. These tasks include: relation extraction, question-answering and recommender systems.

To investigate the effect of several models, the authors have trained the KGE models under an open world assumption (OWA) and tested these models on efficiency and performance on downstream tasks. OWA assumes that the links present in a KG are true, and that links absent in the KG may be true. Models which transform KGs into vectors are concluded to be most efficient. These models are characterised by a smaller space and time complexity than the models which transform KGs into matrices. Against expectations, the models with smaller space and time complexity did not perform significantly worse than the more costly models. The authors explain that this could be an overfitting issue caused by the smaller size of the datasets used.

Lastly, different types of additional information that could be included in the knowledge embedding models are described. These types include: entity types, relation paths, textual descriptions and the implementation of logical rules to derive additional information.

Nickel et al. (2015) review different statistical models that can be trained to predict new relations and new information about objects in KGs. These objects are entities, which are linked to other entities through edges. These edges represent relations between entities, and properties of these entities. Such combinations of two entities and an edge between them are called triples. KGs provide information that can be interpreted by computers. New information in the KG is predicted based on already existing information present in the KG. Nickel et al. (2015) review multiple models.

Latent feature models infer new information based on knowledge that is not explicitly stated in the data. The relationships between entities can be inferred from interactions of their latent features. These interactions can be shaped by different ways of modelling. Nickel et al. (2015) mention several models: RESCAL (bilinear model), Tensor factorization models, Matrix factorization

models, Multi-layer perception models, Neural tensor networks and Latent distance models. The results indicate that the performance of each model is dataset dependent. Latent feature models are best used to infer relational patterns in the whole KG from new latent variables. The computational cost can be lowered by reducing the number of latent variables.

Graph feature models infer new information based on observable information in the KG. The authors also discuss several graph feature models: Similarity measures for uni-relational data, Rule Mining and Inductive Logic Programming and Path Ranking Algorithm. Graph feature models are best used to derive local patterns. The computational cost of graph feature models can be decreased if relations between entities can be explained from the distance of entities.

Furthermore, the authors consider the combination of latent feature models with graph feature models. They state that the combination of both models can yield better results as the authors believe that the models complement each other to increase predictive performance. The models can be combined by adding one model to the other or by collecting separate outputs from both models and combining these outputs as an input for another system.

Moreover, the authors describe different elements that should be considered while training the aforementioned models. The authors indicate that implementing negative samples in your model can be a demanding task. They define different models to cope with this problem, such as creating a local-closed world assumption or generating potentially false edges from text extraction methods.

Lastly, Markov Random Fields are considered. Markov Random Fields assume that triples are conditionally dependent. This means that every triple could be dependent on every other edge and entity. This can lead to scalability issues. However, the number of dependencies can be scaled down by only considering a part of them. The authors conclude that Markov Random Fields are maleable, but that they are more difficult to use for scalable inference than latent feature models and graph feature models.

Training knowledge embedding models relies on the positive as well as the negative links between entities (Kotnis & Nastase, 2017). These positive links represent the presence of a relation between two entities. Negative links represent the absence of a relation between two entities. Kotnis and Nastase (2017) investigate the impact of negative links on knowledge embedding models. The authors test the effect of negative links on different LP models by using different negative sampling methods to train the knowledge embedding models. Kotnis and Nastase (2017) describe that embedding based negative sampling is best used for KGs suffering from data inadequacy. In contrast, for data rich KGs, sampling from corruption is stated to be the best option. The effect of the negative sampling method is dependent on the relation types embedded in these KGs.

2.2 Link Prediction tasks

To perform computations on KGEs (KGE), a framework is needed. PyKEEN is a python based framework which can carry out such computations (Ali et al.,

2019). Ali et al. (2019) perform a LP task on a biomedical KG using PyKEEN. PyKEEN can be used to train and evaluate KGEs's. Using PyKEEN for LP tasks, a score for every possible link can be generated. This score represents the likelihood that this link is true. The authors claim that PyKEEN can be used in any domain, which gives PyKEEN a high usability.

Pujari and Kanawati (2012) propose a rank aggregation approach for LP in complex network. They describe rank aggregation as the difference between two rankings. They mention the Spearman Footrue distance, which computes the distance as the difference between the rankings of each element, and the Kendal Tau distance, which compares two lists and finds the number of disagreements as distance metrics to determine the difference between two rankings.

The authors investigate the implementation of adding weights to the rankers to increase performance of the rank aggregation methods. The authors add weights to both Borda's method and local Kemeny optimal method. Furthermore, multiple ways to calculate these weights are introduced and described. They evaluate their models using rankings of attribute values of the KG. The authors claim that their approach is promising and that it seems to outperform classical machine learning algorithms for LP.

LP is realized by comparing nodes in a KG. This comparison often comes in the form of structural similarity (Zhang et al., 2018). Zhang et al. (2018) review 18 similarity metrics and observe their performance on link prediction and spurious link elimination. The performance of the 18 similarity algorithms is evaluated with the area under the receiver operating characteristic curve or AUC.

The authors have found that algorithms with high predictive accuracy perform badly on spurious link identification. Moreover, the algorithms seem to be impacted by noise. Noise impacts the performance of LP tasks more than the performance of spurious link identification. Moreover, an index is created which depicts the correlation between the AUC of the algorithm on LP and the AUC of the algorithm on spurious link identification. The results show that the algorithms perform on average more stable for spurious link identification than for LP. In addition, the authors state that the 18 algorithms can be categorized into three classes, namely: node-based similarity algorithms, path-based similarity algorithms and Bayesian similarity estimation algorithms.

2.3 Interestingness

Link interestingness is an important aspect of LP tasks (Pusala et al., 2017). Although interestingness measures have been studied widely in other fields such as data mining, Pusala et al. (2017) claim that, to the best of their knowledge, interestingness measures are not yet used to identify the importance of predicted links in LP tasks. Therefore, the authors introduce a rank based approach to rank predicted links based on their importance. The authors analyze thirteen different interestingness measures and state that their approach can predict the rank of future predicted links more precisely than predicting the score of an unknown link. In contrast to aforementioned related work, this paper investigates

link interestingness as a measure based on link prediction scores resulting from overesampling triples. In this work, the effect of positive and negative oversampling are emphasized as an evaluation metric for the proposed link interestingness measure. To the best of my knowledge, this approach is not yet experimented with and should thus contribute to research into link interestingness.

3 Methodology

The main objective of this paper is to develop and evaluate a method to measure link interestingness. To achieve this objective, the method presented in this paper computes an estimated impact that a predicted link has on the information gain of the KG. This section explains the approach, as well as important factors that were taken into consideration.

3.1 Experimental setup

This research experiment is focused on proposing and investigating a link interestingness measure for KGs. This measure is constructed based on the impact that certain triples have on the information gain in the KG. The impact is computed by measuring the difference in rankings of a LP task under the assumption that a triple is positive or negative compared to the situation where the plausibility of this triple is unknown.

In this research, I perform my experiments onto a subset of the Freebase 15k (FB15K) dataset and onto a subset of the WordNet18 (WN18) dataset. The FB15k contains 592,213 triples formed from 14,951 entities and 1,345 relations and was first introduced by Bordes et al. (2013). Moreover, this dataset is well-known in the field of KGEs and is believed to be a benchmark dataset (M. Wang et al., 2021). However, Akrami et al. (2018) discovered that the inverse of triples in the training set corresponded with triples in the testing set. To prevent this problem, I use the FB15k-237 dataset, which is a subset of FB15k, where the inverse duplicate relations are removed. The WN18 dataset contains 141,442 triples with 18 relations scraped from WordNet and was also introduced by Bordes et al. (2013). Similar to the FB15K dataset, inverse relations from the training set resulted in a large amount of triples in the testing set. Therefore, I use the WN18RR dataset instead, which addresses this issue. Relevant KG statistics for these two datasets are shown in Table 1.

Dataset	Fb15k237	WN18RR
number of entities	14505	40559
number of relations	237	11

Table 1. KG statistics for each dataset

The proposed approach has a time complexity of $O(n^3)$, as each pair of entities in the dataset needs to be tested with all relations in the dataset. Especially

with large datasets, the computational costs scale up relatively quickly. I suggest the following technique to address this problem. The entities are first being clustered using a K-medoids algorithm. The number of clusters to be formed was derived from the "elbow" method. The results from the K-elbow algorithm are shown in Figure 1 and Figure 2. For each of these clusters, the medoid is taken which serves as a representative entity for that cluster. From these medoids, all possible combinations of medoids are derived. These entity pairs are used as head-tail combinations, which act as representative head-tail combinations.

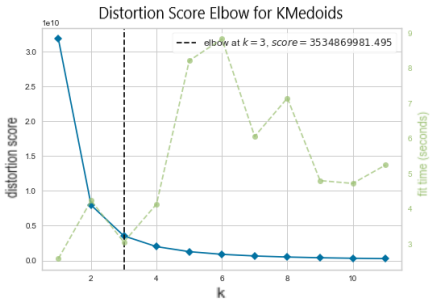

Fig. 1. Distortion score for Kmedoids clustering (FB15k237)

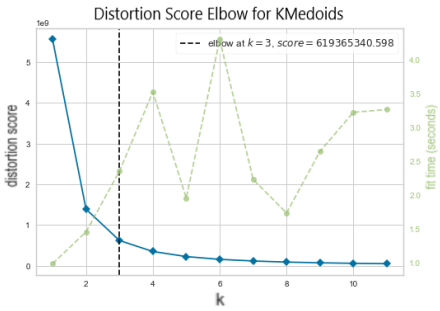

Fig. 2. Distortion score for Kmedoids clustering (WN18RR)

After clustering, the KGE model is trained and a LP task is performed. This situation will be referred to as situation S. In this situation, it is unknown whether a potential triple is true or false. The LP task is performed on the representative head-tail combinations. From the LP task, a list with scores is gathered in which probabilities for all relations between the head-tail combinations are shown. From this list, triples that were not present in the training set are collected. Between the representative head-tail combinations, a number of relations is inserted to create triples. As the WN18RR dataset only contains 11 relations, 11 relations are inserted for both datasets. For the FB15k237 dataset, these are 11 random relations. The resulting triples should be triples that were not yet present in the training set. These triples represent links that are not yet known, which makes them relevant to inspect. Each of these triple are then positively and negatively oversampled, which will be referred to as situation A and situation B, respectively. In both situation A and B, a LP task is performed on the representative head-tail combinations. The interestingness measure is calculated by comparing the scores from situation A and B to situation S. A diagram showing this process is provided in Appendix A. The difference in rankings is computed by the Kendall (Tau's) rank correlation coefficient. Kendall Tau's is a non-parametric test which provides a score for the correlation between two rankings (Brandenburg et al., 2013). The difference in scores should provide an indication of the impact of a triple, where a large difference indicates a higher

impact. The interestingness measure is then defined as follows:

$$E_{int}(h,r,t) = P(A)\frac{-1}{\tau(S,A)} + P(B)\frac{-1}{\tau(S,B)} \tag{1}$$

where E_{int} represents the expected interestingness, A represents the situation in which the triple is assumed to be true, B represents the situation in which the triple is assumed to be false and S represents the situation in which it is unknown whether the triple is true or false. The τ value represents the correspondence between the scores resulting from the LP task. Furthermore, the τ value is a value between 1 and -1, where a value close to 1 represents a strong agreement and values close to -1 a strong disagreement. Therefore, a large expected interestingness indicates a higher impact on the information gain and a smaller expected interestingess indicated a lower impact on the information gain. The proposed model is evaluated on the distribution of scores resulting from the LP task in situation S, A and B. I perform a z-test to determine the impact of oversampling on the distribution of scores.

The model is built on a pipeline function built-in PyKEEN. In the PyKEEN pipeline different variables can be set to train the embedding model. These variables include: *dataset, KGE model, number of epochs, loss function, training loop, negative sampling method and learning rate*. Next, these variables and their values are explained and justified.

To embed the KG into a vector space representation, the RESCAL relation learning aproach is used. RESCAL was first introduced by Nickel et al. (2011). RESCAL is a latent factor model that takes the structure of the KG into account to create tensors. Tensors are multidimensional arrays which describe relationships related to vector spaces (Kolda & Bader, 2009). RESCAL suffers from overfitting due to the large amount of parameters. (Kong et al., 2019) The scoring function for RESCAL is defined as follows:

$$f(h,r,t) = \mathbf{e}_h^T \mathbf{W}_r \mathbf{e}_t = \sum_{i=1}^{d}\sum_{j=1}^{d} w_{ij}^{(r)} (\mathbf{e}_h)_i (\mathbf{e}_t)_j \tag{2}$$

As Nickel et al. (2015) describe in their review of relation machine learning models, training under the open world assumption (OWA) leads to underfitting. Under the open world assumption, triples that are not yet present in the KG are considered to be obscure. OWA assumes that the relation of this triple could be true and it could be false. For this research experiment, I use a training loop under the closed world assumption (CWA).

PyKEEN provides two CWA training loops to train the embedding model, a local closed world assumption (LCWA) training loop and a stochastic local closed world assumption (sLCWA) training loop. For this particular experiment, the sLCWA training loop is used. The sLCWA method is used as it directly impacts oversampling and therefore this research experiment. With a sLCWA, triples can be assumed to be true or false based on their presence in the KG. The sLCWA training loop also comes with advantages over the LCWA training loop. The

LCWA training loop considers triples that are not present in the KG as negative triples, based on one of three strategies. These strategies are categorized in a head, relation and tail strategy, in which the negative triples are generated based on the presence of either the head or tail in the set of entities and the presence of a relation in the set of relations. The sLCWA takes a random subset of the union of the head and tail strategies of the LCWA training loop. The sLCWA method reduces the computational workload, affects a small set of the KGE and it creates the opportunity to develop novel negative sampling strategies.

As a loss function, I use the binary cross entropy loss (BCE With logits Loss). This loss function is formulated as follows:

$$L(h,r,t) = -(l(h,r,t)\cdot\log(\sigma(f(h,r,t)))+(1-l(h,r,t))\cdot\log(1-\sigma(f(h,r,t)))) \quad (3)$$

where $\sigma(x)$ represents the sigmoid function:

$$\frac{1}{1+\exp(-x)} \quad (4)$$

and where f represents the interaction function $f : \mathcal{E} \times \mathcal{R} \times \mathcal{E} \to R$ and where l represents the label function $l : \mathcal{E} \times \mathcal{R} \times \mathcal{E} \to \{0, 1\}$, where $\mathcal{E}$ represents the set of entities and $\mathcal{R}$ represents the set of relations. In contrary to MarginRankingLoss, the BCE With logits Loss considers absolute values of the scores instead of only the relative difference in scores. Furthermore, the cross entropy loss function can be used to show the LP scores as the result of a sigmoid function. This is needed to gather the right probabilities that are used in the interestingness measure.

The number of epochs set, which refers to the amount of training loops the model trains on, is different per dataset. As Figures 3 and 4 show, the loss converges to 0 as the amount of epochs increases. Therefore, I choose the number of epochs based on the epoch after which the difference in losses is smaller than $1e^{-}06$. The number of epochs are 13 and 47 for the FB15k-237 and the WN18RR dataset, respectively. With these number of epochs, the model should be trained sufficiently to yield significant scores.

To create the assumption that a triple is positive or negative, the triple is both positively and negatively oversampled. Z. Yang et al. (2020) express the importance of negative sampling in training Knowledge Embedding Models. They prove that negative sampling is as important as positive sampling. The set of triples that are not present in the KG are labeled as negative under the closed world assumption. This implies that there are significantly more potential negative triples than positive triples. PyKEEN comes with a basic negative sampling strategy which uniformly randomly corrupts triples using either of two operations. These operations either corrupt the head or the tail of a triple. Table 2 shows these operations.

In the situation where the triple is assumed to be positive, the basic negative sampling strategy is used. On the contrary, in the situation where the triple is assumed to be negative, the triple is negatively oversampled using a new negative sampling strategy. This new negative sampling strategy is used to negatively oversample specific triples instead of randomly uniformly corrupting triples. This

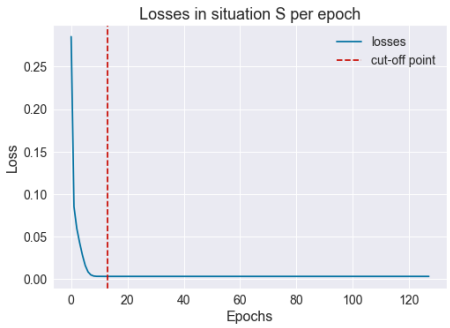

Fig. 3. Losses in situation S per epoch (FB15k237)

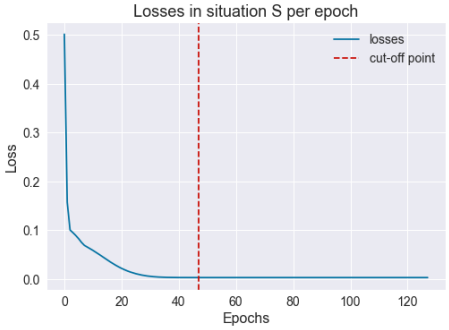

Fig. 4. Losses in situation S per epoch (WN18RR)

Corrupt Heads	$\mathcal{H}(h,r,t) = \{(h',r,t)\|h' \in \mathcal{E} \wedge h' \neq h\}$
Corrupt Tails	$\mathcal{T}(h,r,t) = \{(h',r,t)\|t' \in \mathcal{E} \wedge t' \neq t\}$

Table 2. Corruption operations

is needed to assume that a specific triple is negative (situation B). The new negative sampling strategy builds on the basic negative sampling strategy. In addition to uniformly randomly corrupting triples, the new negative sampling strategy replaces a portion of the corrupted set with the to be oversampled triple.

In addition, I experimented with multiple learning rates to find that the learning rate $\lambda = 0.0001$ performs best. Table 3 shows the variance in scores for each learning rate that I experimented with. 3 shows that there is a local optimum for the learning rate. Although the variance is still very low for $\lambda = 0.0001$, the variance for $\lambda = 0.0001$ is significantly higher. Important to note is that the variance for $\lambda = 0.0001$ is higher for the FB15k237 dataset than for the WN18RR dataset. This is largely due to the size of the relation set.

	Learning Rate λ		
Dataset	0.001	0.0001	0.00001
FB15k237	0.000	0.012396	0.000
WN18RR	0.000	0.00085	0.000

Table 3. Variance in LP scores in situation S for each learning rate

4 Findings

This section concerns the main findings of this work. The proposed interestingness measure is evaluated on the impact of both positive and negative oversampling. The impact of positive and negative oversampling is measured by performing a two-sampled z-test. Z-test approximates whether two sample means are the same or different and is defined by formula 5, where $\overline{X_1}$ and $\overline{X_2}$ represent the sample mean, μ_1 and μ_1 the population mean, σ_1 and σ_2 the standard deviation of the population and n_1 and n_2 the sample size corresponding to the LP scores resulting from situation S and A. The z-test is performed on the scores resulting from the LP task in situation S and the scores resulting the LP task after oversampling a triple. This implies that the impact is measured for every oversampled triple. For each oversampled triple, the expected interestingness is calculated. To further evaluate the model, the triple with the highest expected interestingness for each amount of oversampling is examined. This triple is evaluated on the impact that this triple has on the LP scores.

$$Z = \frac{(\overline{X_1} - \overline{X_1}) - (\mu_1 - \mu_2)}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}} \tag{5}$$

Using the Z-test statistic, I test the following null hypothesis and alternative hypothesis against each other for each oversampled triple.

$$H_0 : \mu_1 - \mu_2 = 0 \tag{6}$$

$$H_1 : \mu_1 - \mu_2 \neq 0 \tag{7}$$

Using the Z-test statistic, I test the following null hypothesis and alternative hypothesis against each other for the triples with the highest expected interestingness.

$$H_0 : \mu_1 - \mu_2 = 0 \tag{8}$$

$$H_1 : \mu_1 - \mu_2 > 0 \tag{9}$$

If the resulting p-value < 0.05, the null hypothesis is rejected. For each oversampled triple, this means that there is sufficient evidence to conclude that the two sample means from the distribution of LP scores are different. For the triples with the highest expected interestingness, this means that there is sufficient evidence to conclude that the sample mean from the LP scores are higher when adding this triple with the highest expected interestingness. This indicates that oversampling has an impact on the resulting LP scores. In addition, the average variance of the distributions of all triples is measured. The distribution of the scores resulting from the LP task in situation S are shown in Figure 5 and Figure 6.

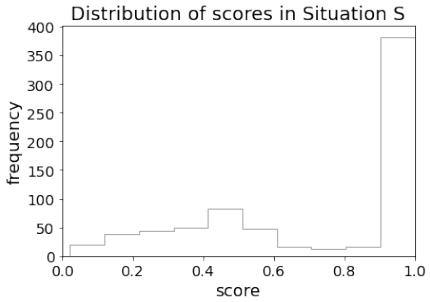

Fig. 5. Distribution of scores in situation S (FB15k237)

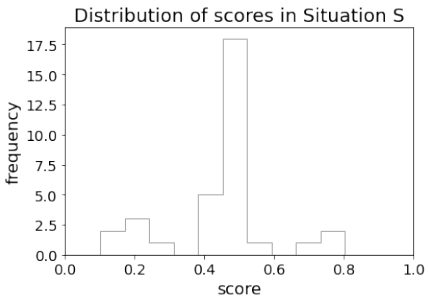

Fig. 6. Distribution of scores in situation S (WN18RR)

4.1 Positive Oversampling

The results from the z-test for positive oversampling are summarized in Table 4. Furthermore, Table 4 displays the average variance of the distributions resulting from the LP task in situation A. The columns are divided into three parts, which correspond to three different amounts of oversampling. These percentages show how many triples, relative to the amount of triples present in the training set in situation S, are added in situation A. For each part, the average variance and the percentage of p-values where the p-values < 0.05 are shown. A full list of all the p-values for each triple are shown in Table 8, 9, 10 and 11 in appendices B, C, D and E.

	FB15k237			WN18RR		
	0.01%	0.1%	1%	0.01%	0.1%	1%
avg. variance	0.094	0.068	0.001	0.013	0.001	0.000
p-value < 0.05	0.0%	0.0%	100%	0.0%	0.0%	0.0%

Table 4. Effect of positive oversampling on the distribution of scores for each percentage of oversampling

The results show that the average variance decreases for a higher percentage of oversampling for the FB15k237 dataset. Similar to the average variances of the FB15k237, for the WN18RR dataset, the average variance decreases with a higher percentage of oversampling. This suggests that positive oversampling has an impact on the distribution of LP scores. Table 4 furthermore shows that for 1% oversampling the p-values < 0.05 for the FB15k237 dataset. Therefore, for 1% oversampling for the FB15k237 dataset, I reject the null hypothesis. This indicates that 1% positive oversampling for the FB15k237 dataset is correlated with impacting the distribution of LP scores. All other scenarios for positive oversampling resulted in p-values larger than 0.05. Therefore, in these scenarios,

the null hypothesis is not rejected. This indicates that 0.01% and 0.1% positive oversampling are not correlated with impacting the LP scores for the FB15k237 dataset. This implies that larger amounts of positive oversampling impact LP scores, which suggests that the proposed interestingness measure should yield significant results for the appropriate amount of oversampling on the FB15k237 dataset. Moreover, it is shown that 0.01%, 0.1% and 1% oversampling are not correlated with impacting the LP scores for the WN18RR dataset. This indicates that the proposed interestingness measure should not yield significant results for the WN18RR dataset for 0.01%, 0.1% and 1% positive oversampling.

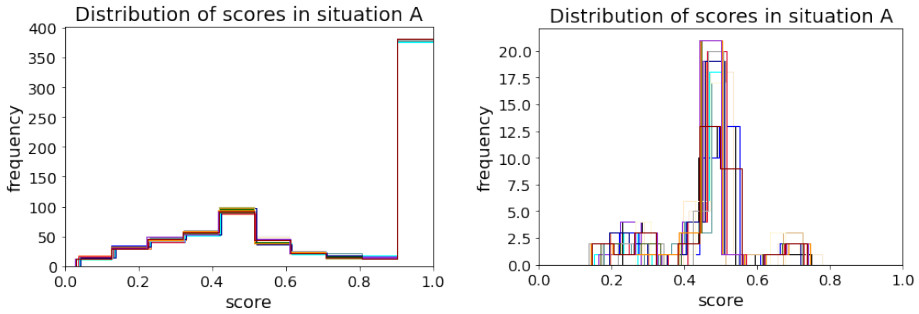

Fig. 7. Distribution of scores in situation A with 0.01% positive oversampling with from left to right the FB15k237 and the WN18RR dataset

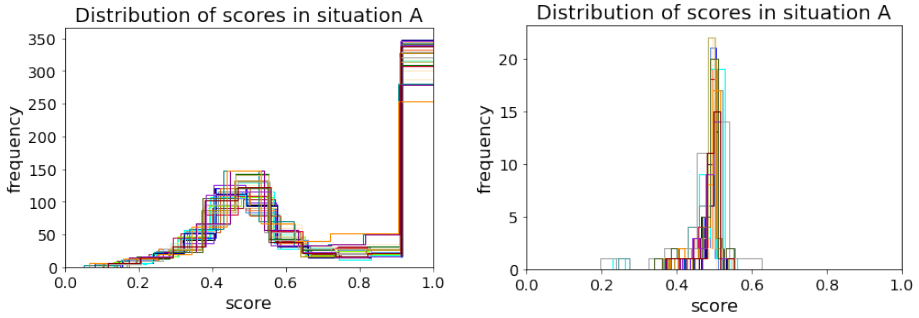

Fig. 8. Distribution of scores in situation A with 0.1% positive oversampling with from left to right the FB15k237 and the WN18RR dataset

Figure 7, Figure 8 and Figure 9 display the distributions of LP scores in situation A for each oversampled triple for each percentage of oversampling. These figures show that 0.01% and 0.1% positive oversampling have a relatively small impact on the distribution of LP scores, whereas 1% positive oversampling

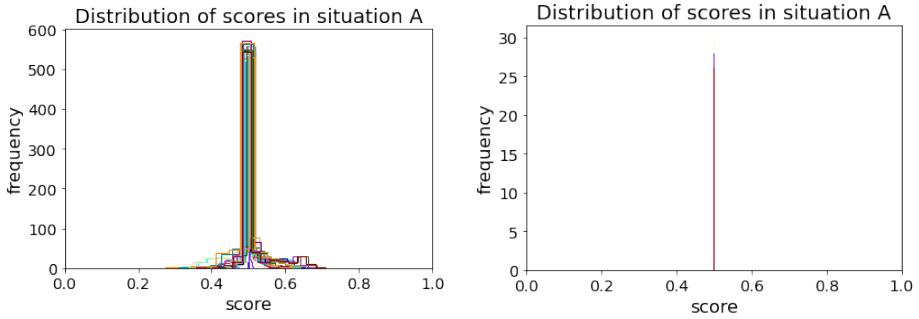

Fig. 9. Distribution of scores in situation A with 1% positive oversampling with from left to right the FB15k237 and the WN18RR dataset

has a relatively large impact on the distribution of LP scores. These results are as expected. In addition, these figures show that for an increasing amount of positive oversampling, the LP scores tend to fluctuate more around 0.5. Moreover, the figures show that positive oversampling also has an impact on the distribution of LP scores for the WN18RR dataset, which was not seen in the z-test. This implies that for both datasets, the proposed interestingness measure should yield significant results for the appropriate amount of positive oversampling.

	FB15k237			WN18RR		
	0.01%	0.1%	1%	0.01%	0.1%	1%
expected interestingness	-1.005	-1.020	-1.091	-1.022	-1.027	-1.053
z-score	0.376	0.503	-19.514	0.505	1.423	1.750
p-value	0.354	0.308	1.0	0.307	0.077	0.040
in test set	False	False	False	False	False	False

Table 5. Effect of positive oversampling on the LP scores for each percentage of oversampling for the triple with highest interestingness

The effect of positive oversampling for the triples with the highest expected interestingness is presented in Table 5. Table 5 shows that the expected interestingness is relatively low. Furthermore, for the FB15k237 dataset with 1% oversampling, the z-score is negative, which implies that the mean LP score is lower when the triple with the highest expected interestingness is added. However, for 0.1% and 0.01% oversampling, the z-scores are positive, which implies that the mean LP score is higher when the triple with the highest expected interestingness is added. For the WN18RR dataset, this is the case for each amount of oversampling. This suggests that the effect of positive oversampling is dataset dependent. For the WN18RR dataset with 1% oversampling, the p-value < 0.05. Therefore, for the WN18RR dataset with 1% oversampling, I reject the null hy-

pothesis. This indicates that the LP scores for 1% positive oversampling are significantly higher when the triple with the highest expected interestingness is added to the WN18RR KG. Table 5 furthermore shows that the triples with the highest expected interestingness were not in the testing set. Therefore I assume that these triples are actually negative triples.

4.2 Negative Oversampling

The results from the z-test for negative oversampling are summarized in Table 6. The results show that the average variance decreases for a higher percentage of oversampling for both datasets. Although the variances for 0.01% and 0.1% oversampling are equal, the variance for 1% oversampling is near zero. Table 6 further shows that for 1% oversampling the p-values < 0.05 for the FB15k237 dataset. Therefore, for 1% oversampling for the FB15k237 dataset, I reject the null hypothesis. This indicates that 1% negative oversampling for the FB15k237 dataset is correlated with impacting the distribution of LP scores. All other scenarios for negative oversampling resulted in p-values larger than 0.05. Therefore, in these scenarios, the null hypothesis is not rejected. This indicates that 0.01% and 0.1% negative oversampling are not correlated with impacting the LP scores for the FB15k237 dataset. This implies that larger amounts of negative oversampling impact LP scores, which suggests that the proposed interestingness measure should yield significant results for the appropriate amount of oversampling on the FB15k237 dataset Moreover it is implied that 0.01%, 0.1% and 1% oversampling are not correlated with impacting the LP scores for the WN18RR dataset. This indicates that the proposed interestingness measure should not yield significant results for the WN18RR dataset for 0.01%, 0.1% and 1% negative oversampling.

	FB15k237			WN18RR		
	0.01%	0.1%	1%	0.01%	0.1%	1%
avg. variance	0.100	0.100	0.001	0.024	0.024	0.000
p-value < 0.05	0.0%	0.0%	100%	0.0%	0.0%	0.0%

Table 6. Effect of negative oversampling on the distribution of scores for each percentage of oversampling

Figure 10, Figure 11 and Figure 12 display the distributions of LP scores in situation B for each oversampled triple for each percentage of oversampling. These figures show that 0.01% and 0.1% negative oversampling have a near zero impact on the distribution of LP scores, whereas 1% negative oversampling has a significant impact on the distribution of LP scores. Moreover, the figures show that negative oversampling has an impact on the distribution of LP scores for both datasets. This implies that for both datasets, the proposed interestingness

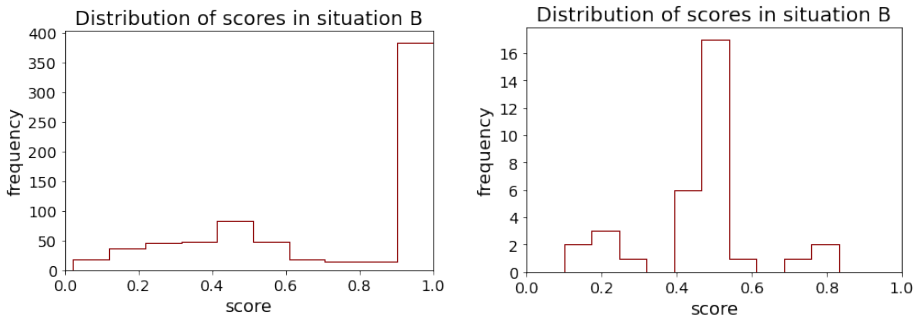

Fig. 10. Distribution of scores in situation A with 0.01% negative oversampling with from left to right the FB15k237 and the WN18RR dataset

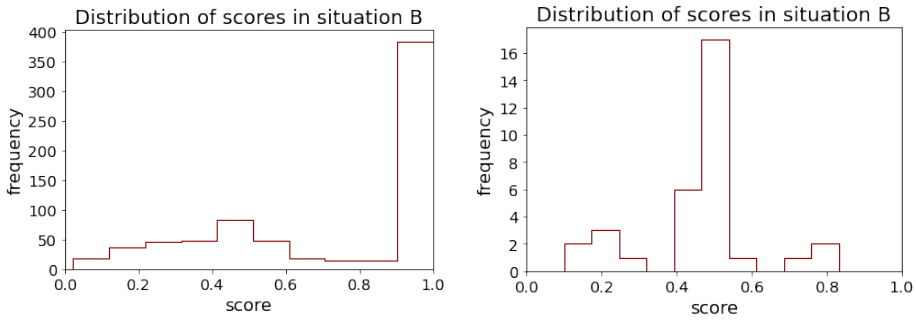

Fig. 11. Distribution of scores in situation A with 0.1% negative oversampling with from left to right the FB15k237 and the WN18RR dataset

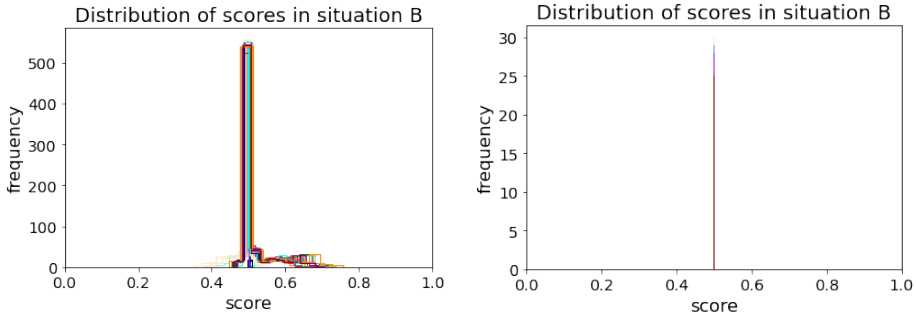

Fig. 12. Distribution of scores in situation A with 1% negative oversampling with from left to right the FB15k237 and the WN18RR dataset

measure should yield significant results for the appropriate amount of negative oversampling.

	FB15k237			WN18RR		
	0.01%	0.1%	1%	0.01%	0.1%	1%
expected interestingness	-1.020	-1.020	-1.142	-1.027	-1.027	-1.258
z-score	0.054	0.054	-18.567	0.100	0.100	1.750
p-value	0.478	0.478	1.0	0.460	0.460	0.040
in test set	False	False	False	False	False	False

Table 7. Effect of negative oversampling on the LP scores for each percentage of oversampling for the triple with highest interestingness

The effect of negative oversampling for the triples with the highest expected interestingness is presented in Table 7. Similar to Table 5, Table 7 shows that the expected interestingness is relatively low. Furthermore, for the FB15k237 dataset with 1% negative oversampling, the z-score is negative, which implies that the mean LP score is lower when the triple with the highest expected interestingness is added. However, for 0.1% and 0.01% oversampling, the z-scores are positive, which implies that the mean LP score is higher when the triple with the highest expected interestingness is added. For the WN18RR dataset, this is the case for each amount of oversampling. This suggests that the effect of negative oversampling is dataset dependent. For the WN18RR dataset with 1% oversampling, the p-value < 0.05. Therefore, for the WN18RR dataset with 1% oversampling, I reject the null hypothesis. This indicates that the LP scores for 1% positive oversampling are significantly higher when the triple with the highest expected interestingness is added to the WN18RR KG. Table 7 furthermore shows that the triples with the highest expected interestingness were not in the testing set. Therefore I assume that these triples are actually negative triples. Additionally, the z-scores and corresponding p-values of 0.01% and 0.1% negative oversampling are equal for each dataset. This indicates that the effect of negative oversampling does not decrease anymore after 0.1% negative oversampling with decreasing amounts of negative oversampling.

5 Discussion

This paper has investigated a possible approach to measure link interestingness in KGs. The proposed model measures link interestingness by comparing LP scores resulting from both positively and negatively oversampling triples.

The results show that an increasing amount of positive oversampling for the FB15k237 dataset results in a decrease of variance, but an increase in impact on the distribution of LP scores. The decrease in variance could possible be explained by the uncertainty of the model. As the amount of positive oversampling increases, it seems that the distribution of LP scores becomes more concentrated around 0.5. This indicates that the model becomes more uncertain about other triples when oversampling a specific triple. A reduction in variance emphasizes the impact of positive oversampling.

In addition, the results show that an increasing amount of negative oversampling results in a reduction of variance. While the average variances are equal for 0.01% and 0.1% oversampling, the average variance drastically decreases to near zero for 1% oversampling. For both datasets, with an increasing amount of oversampling, scores fluctuate around 0.5. However, for the WN18RR dataset, scores already fluctuated around 0.5. This could explain why the z-test does not imply a difference in distributions.

It is to be noted that the amount of LP scores is relatively small for the WN18RR dataset compared to the amount of LP scores of the FB15k237 dataset. The number of relations present in both datasets plays a central role in this difference. As the WN18RR dataset only has 11 relations, less scores are collected from the LP task, making the distribution of the WN18RR dataset smaller than the distribution of the FB15k237 dataset.

Moreover, for the distributions of the FB15k237 dataset, depicted in Figure 5, a peak of frequencies can be seen for scores between 0.9 and 1.0. This could possibly be explained by the LP task. The LP task predicts probability scores for all relations between the representative head-tail combinations. There is a probability that some scores were scores for triples that were already in the training set. Thus, these scores naturally tend to scores close to 1.

The evaluation of the model was mainly conducted by investigating the impact of positive and negative oversampling, as these variables play a central role in the proposed interestingness model. However, there are more factors that could be taken into consideration. Without clustering, this model will still be computationally expensive since every possible triple should be investigated/oversampled to understand their interestingness and to rank all potential triples. For this research experiment, the computational costs were lowered by clustering the entities. To identify interestingness of all links in a KG in a real life setting, all potential triples should be inspected. This would be computationally very expensive, especially for large datasets. Though, it could be argued that these computations are still less expensive than experimentally testing potential triples. Also, the proposed clustering of entities should solve this problem to some extent. However, in biomedical settings a high precision is essential. Therefore, in these settings, clustering can help to identify relevant clusters to investigate, but all potential triples in that cluster should still be ranked by their interestingness.

The results furthermore show that the triples with the highest expected interestingness does not significantly increase the LP scores after adding those triples to the KG for the FB15k237 dataset. The results suggest that adding the triples with the highest expected interestingness does have some effect on the LP scores, but this effect is not significant enough. For the WN18RR dataset, the results show that larger amounts of oversampling result in a significant increase of the LP scores when adding the triples with the highest expected interestingness to the KG. This implies that the impact of oversampling is dataset dependent.

6 Conclusion

This paper has evaluated a proposed method for measuring link interestingness in KGs. This research experiment suggests that positive and negative oversampling has an impact on the distribution of LP scores for higher percentages of oversampling. Adding triples with the highest expected interestingness does not result in a significant increase of LP performance for the FB15k237 dataset. However, for the WN18RR dataset, adding these triples does result in a significant increase of LP performance. This implies that the effect of oversampling is dataset dependent. Regardless, it can be stated that the proposed interestingness measure is a reasonable approach to measure link interestingness to some extent. Nevertheless, the interestingness measure can still be analyzed further. This paper has proposed an outset for measuring link interestingness, on which could be built further to increase the truthfulness of the model.

7 Future work

Further research should focus on testing different settings and values for the KGE model. The impact of variables such as the embedding dimension can be further investigated. Moreover, the parameters of the PyKEEN pipeline such as the optimizer or evaluator could be further optimized for the KGE model. In addition, other loss functions such as Margin Ranking loss could be investigated. This research experiment has experimented with a default pipeline for which to some extent, parameters were optimized. However, these parameters were optimized such that the model would show understandable results. Future research could investigate on optimizing the embedding parameters, after which the link interestingness measure could be tested.

Moreover, future research could examine the impact of different embedding models on the proposed interestingness measure. For this experiment, RESCAL was used as an embedding model, but other embedding models could be explored as well. Potentially, this could lead to a further understanding of defining a link interestingness measure.

Acknowledgement

This work was carried out on the Dutch national e-infrastructure with the support of SURF Cooperative.

A Appendix A

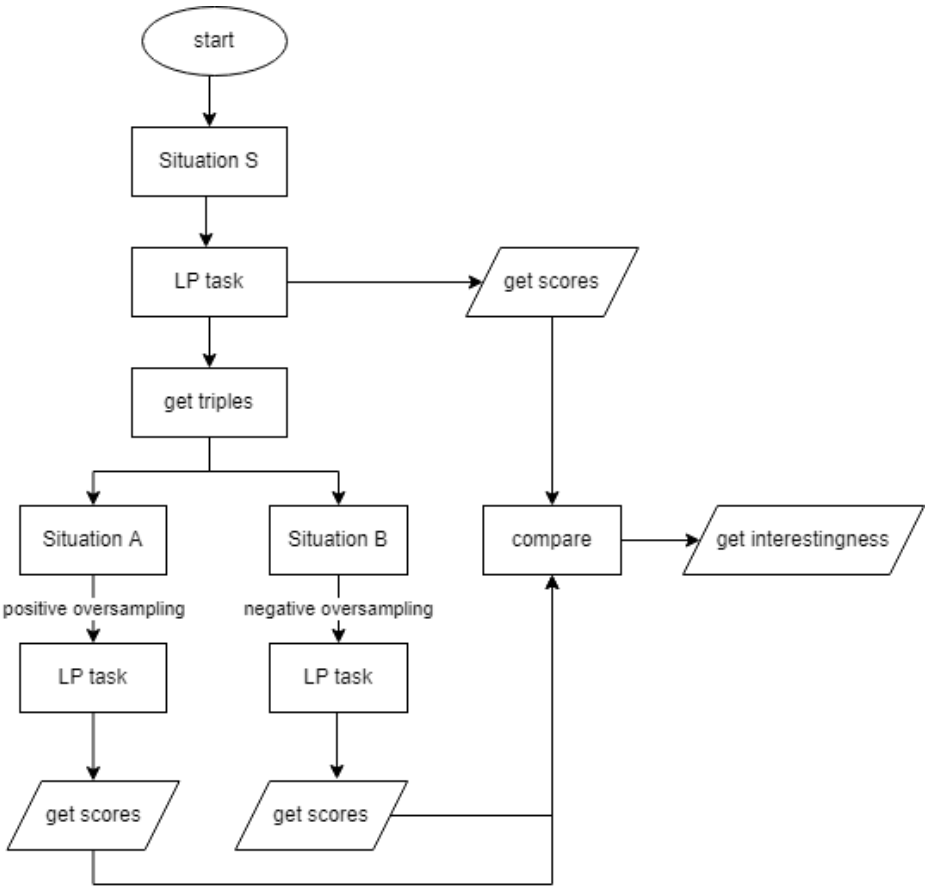

Fig. 13. Diagram showing the global outlines of the proposed approach

B Appendix B

Triple	0.01%			0.1%			1%		
	variance	z-score	p-value	variance	z-score	p-value	variance	z-score	p-value
1	0.093	-0.39	0.698	0.065	0.53	0.597	0.001	20.48	0.000
2	0.094	-0.35	0.723	0.069	-0.30	0.766	0.001	20.63	0.000
3	0.093	-0.32	0.748	0.068	-0.16	0.870	0.001	20.53	0.000
4	0.093	-0.38	0.706	0.067	0.20	0.839	0.001	20.56	0.000
5	0.093	-0.39	0.694	0.068	-0.03	0.980	0.000	20.04	0.000
6	0.093	-0.39	0.697	0.067	0.13	0.897	0.001	20.36	0.000
7	0.093	-0.39	0.695	0.065	0.56	0.574	0.001	20.24	0.000
8	0.094	-0.35	0.725	0.070	-0.53	0.599	0.000	19.59	0.000
9	0.094	-0.39	0.699	0.064	0.95	0.344	0.002	20.71	0.000
10	0.094	-0.35	0.726	0.070	-0.53	0.598	0.000	19.59	0.000
11	0.093	-0.38	0.707	0.069	-0.50	0.615	0.000	19.51	0.000
12	0.094	-0.37	0.711	0.069	-0.19	0.848	0.000	19.84	0.000
13	0.094	-0.33	0.739	0.070	-0.71	0.479	0.001	18.66	0.000
14	0.094	-0.29	0.773	0.071	-0.55	0.580	0.000	19.23	0.000
15	0.094	-0.36	0.718	0.070	-0.41	0.680	0.000	19.32	0.000
16	0.094	-0.37	0.712	0.071	-0.47	0.636	0.001	18.46	0.000
17	0.094	-0.36	0.721	0.070	-0.41	0.685	0.000	19.29	0.000
18	0.094	-0.35	0.724	0.069	-0.19	0.852	0.000	19.81	0.000
19	0.094	-0.32	0.746	0.072	-0.69	0.488	0.001	18.45	0.000
20	0.095	-0.37	0.713	0.067	0.07	0.948	0.000	19.69	0.000
21	0.094	-0.32	0.749	0.072	-0.71	0.476	0.001	18.56	0.000
22	0.094	-0.36	0.721	0.071	-0.75	0.456	0.001	18.42	0.000
23	0.093	-0.41	0.680	0.064	0.83	0.409	0.000	19.90	0.000
24	0.094	-0.34	0.733	0.068	-0.25	0.804	0.001	18.78	0.000
25	0.093	-0.32	0.750	0.068	-0.17	0.869	0.000	19.15	0.000
26	0.093	-0.39	0.696	0.066	0.22	0.827	0.000	19.12	0.000
27	0.093	-0.40	0.688	0.068	-0.06	0.952	0.001	18.53	0.000
28	0.093	-0.40	0.692	0.066	0.16	0.871	0.000	19.49	0.000
29	0.094	-0.39	0.695	0.064	0.82	0.411	0.000	19.50	0.000
30	0.095	-0.30	0.762	0.071	-0.50	0.617	0.002	17.99	0.000
31	0.094	-0.42	0.678	0.062	1.43	0.151	0.001	19.98	0.000
32	0.095	-0.30	0.762	0.071	-0.53	0.598	0.001	18.22	0.000
33	0.094	-0.34	0.735	0.070	-0.48	0.634	0.002	18.03	0.000

Table 8. Complete list with z-scores and variance for every positively oversampled triple (FB15k237)

C Appendix C

Triple	0.01% variance	0.01% z-score	0.01% p-value	0.1% variance	0.1% z-score	0.1% p-value	1% variance	1% z-score	1% p-value
1	0.100	-0.05	0.96	0.100	-0.05	0.96	0.000	20.22	0.000
2	0.100	-0.05	0.96	0.100	-0.05	0.96	0.000	20.12	0.000
3	0.100	-0.05	0.96	0.100	-0.05	0.96	0.000	19.86	0.000
4	0.100	-0.05	0.96	0.100	-0.05	0.96	0.000	20.05	0.000
5	0.100	-0.05	0.96	0.100	-0.05	0.96	0.001	20.31	0.000
6	0.100	-0.05	0.96	0.100	-0.05	0.96	0.001	20.32	0.000
7	0.100	-0.05	0.96	0.100	-0.05	0.96	0.000	20.19	0.000
8	0.100	-0.05	0.96	0.100	-0.05	0.96	0.000	19.53	0.000
9	0.100	-0.05	0.96	0.100	-0.05	0.96	0.001	20.41	0.000
10	0.100	-0.05	0.96	0.100	-0.05	0.96	0.000	19.56	0.000
11	0.100	-0.05	0.96	0.100	-0.05	0.96	0.000	19.59	0.000
12	0.100	-0.05	0.96	0.100	-0.05	0.96	0.001	18.43	0.000
13	0.100	-0.05	0.96	0.100	-0.05	0.96	0.001	18.32	0.000
14	0.100	-0.05	0.96	0.100	-0.05	0.96	0.002	18.00	0.000
15	0.100	-0.05	0.96	0.100	-0.05	0.96	0.001	18.28	0.000
16	0.100	-0.05	0.96	0.100	-0.05	0.96	0.001	18.57	0.000
17	0.100	-0.05	0.96	0.100	-0.05	0.96	0.001	18.69	0.000
18	0.100	-0.05	0.96	0.100	-0.05	0.96	0.001	18.43	0.000
19	0.100	-0.05	0.96	0.100	-0.05	0.96	0.001	18.46	0.000
20	0.100	-0.05	0.96	0.100	-0.05	0.96	0.001	18.60	0.000
21	0.100	-0.05	0.96	0.100	-0.05	0.96	0.001	18.39	0.000
22	0.100	-0.05	0.96	0.100	-0.05	0.96	0.001	18.41	0.000
23	0.100	-0.05	0.96	0.100	-0.05	0.96	0.002	17.79	0.000
24	0.100	-0.05	0.96	0.100	-0.05	0.96	0.002	18.02	0.000
25	0.100	-0.05	0.96	0.100	-0.05	0.96	0.002	17.81	0.000
26	0.100	-0.05	0.96	0.100	-0.05	0.96	0.002	17.90	0.000
27	0.100	-0.05	0.96	0.100	-0.05	0.96	0.002	18.03	0.000
28	0.100	-0.05	0.96	0.100	-0.05	0.96	0.002	17.73	0.000
29	0.100	-0.05	0.96	0.100	-0.05	0.96	0.002	18.00	0.000
30	0.100	-0.05	0.96	0.100	-0.05	0.96	0.001	18.40	0.000
31	0.100	-0.05	0.96	0.100	-0.05	0.96	0.002	17.51	0.000
32	0.100	-0.05	0.96	0.100	-0.05	0.96	0.001	18.22	0.000
33	0.100	-0.05	0.96	0.100	-0.05	0.96	0.002	18.16	0.000

Table 9. Complete list with z-scores and variance for every negatively oversampled triple (FB15k237)

D Appendix D

Triple	0.01% variance	0.01% z-score	0.01% p-value	0.1% variance	0.1% z-score	0.1% p-value	1% variance	1% z-score	1% p-value
1	0.014	-0.21	3	0.000	-1.36	0.174	0.000	-1.75	0.08
2	0.015	-0.34	3	0.000	-1.35	0.178	0.000	-1.75	0.08
3	0.010	0.68	3	0.003	-0.98	0.328	0.000	-1.75	0.08
4	0.011	-0.44	3	0.001	-1.42	0.155	0.000	-1.75	0.08
5	0.014	-0.21	3	0.000	-1.36	0.174	0.000	-1.75	0.08
6	0.014	-0.24	3	0.000	-1.38	0.166	0.000	-1.75	0.08
7	0.014	-0.24	3	0.000	-1.38	0.166	0.000	-1.75	0.08
8	0.011	-0.44	3	0.001	-1.42	0.155	0.000	-1.75	0.08
9	0.013	-0.36	3	0.000	-1.34	0.180	0.000	-1.75	0.08
10	0.014	-0.21	3	0.001	-1.37	0.169	0.000	-1.75	0.08
11	0.014	-0.21	3	0.000	-1.36	0.174	0.000	-1.75	0.08
12	0.014	-0.23	3	0.001	-1.26	0.207	0.000	-1.75	0.08
13	0.014	-0.25	3	0.003	-1.25	0.210	0.000	-1.75	0.08
14	0.010	0.72	3	0.001	-0.77	0.441	0.000	-1.75	0.08
15	0.009	-0.50	3	0.000	-1.32	0.187	0.000	-1.75	0.08
16	0.014	-0.23	3	0.000	-1.26	0.207	0.000	-1.75	0.08
17	0.014	-0.22	3	0.001	-1.31	0.192	0.000	-1.75	0.08
18	0.014	-0.22	3	0.001	-1.28	0.200	0.000	-1.75	0.08
19	0.009	-0.50	3	0.001	-1.32	0.187	0.000	-1.75	0.08
20	0.013	-0.41	3	0.000	-1.22	0.222	0.000	-1.75	0.08
21	0.014	-0.21	3	0.001	-1.29	0.195	0.000	-1.75	0.08
22	0.014	-0.23	3	0.001	-1.26	0.207	0.000	-1.75	0.08
23	0.015	-0.16	3	0.005	-1.16	0.246	0.000	-1.75	0.08
24	0.016	-0.20	3	0.002	-1.16	0.247	0.000	-1.75	0.08
25	0.011	0.71	3	0.001	-0.74	0.457	0.000	-1.75	0.08
26	0.014	-0.31	3	0.001	-1.10	0.270	0.000	-1.75	0.08
27	0.015	-0.16	3	0.001	-1.16	0.246	0.000	-1.75	0.08
28	0.015	-0.18	3	0.002	-1.20	0.231	0.000	-1.75	0.08
29	0.015	-0.15	3	0.001	-1.17	0.243	0.000	-1.75	0.08
30	0.014	-0.31	3	0.002	-1.10	0.270	0.000	-1.75	0.08
31	0.015	-0.33	3	0.001	-1.14	0.253	0.000	-1.75	0.08
32	0.015	-0.17	3	0.001	-1.16	0.248	0.000	-1.75	0.08
33	0.015	-0.16	3	0.001	-1.16	0.246	0.000	-1.75	0.08

Table 10. Complete list with z-scores and variance for every positively oversampled triple (WN18RR)

E Appendix E

Triple	0.01%			0.1%			1%		
	variance	z-score	p-value	variance	z-score	p-value	variance	z-score	p-value
1	0.024	-0.10	0.921	0.024	-0.10	0.921	0.000	-1.75	0.08
2	0.024	-0.10	0.921	0.024	-0.10	0.921	0.000	-1.75	0.08
3	0.024	-0.10	0.921	0.024	-0.10	0.921	0.000	-1.75	0.08
4	0.024	-0.10	0.921	0.024	-0.10	0.921	0.000	-1.75	0.08
5	0.024	-0.10	0.921	0.024	-0.10	0.921	0.000	-1.75	0.08
6	0.024	-0.10	0.921	0.024	-0.10	0.921	0.000	-1.75	0.08
7	0.024	-0.10	0.921	0.024	-0.10	0.921	0.000	-1.75	0.08
8	0.024	-0.10	0.921	0.024	-0.10	0.921	0.000	-1.75	0.08
9	0.024	-0.10	0.921	0.024	-0.10	0.921	0.000	-1.75	0.08
10	0.024	-0.10	0.921	0.024	-0.10	0.921	0.000	-1.75	0.08
11	0.024	-0.10	0.921	0.024	-0.10	0.921	0.000	-1.75	0.08
12	0.024	-0.10	0.921	0.024	-0.10	0.921	0.000	-1.75	0.08
13	0.024	-0.10	0.921	0.024	-0.10	0.921	0.000	-1.75	0.08
14	0.024	-0.10	0.921	0.024	-0.10	0.921	0.000	-1.75	0.08
15	0.024	-0.10	0.921	0.024	-0.10	0.921	0.000	-1.75	0.08
16	0.024	-0.10	0.921	0.024	-0.10	0.921	0.000	-1.75	0.08
17	0.024	-0.10	0.921	0.024	-0.10	0.921	0.000	-1.75	0.08
18	0.024	-0.10	0.921	0.024	-0.10	0.921	0.000	-1.75	0.08
19	0.024	-0.10	0.921	0.024	-0.10	0.921	0.000	-1.75	0.08
20	0.024	-0.10	0.921	0.024	-0.10	0.921	0.000	-1.75	0.08
21	0.024	-0.10	0.921	0.024	-0.10	0.921	0.000	-1.75	0.08
22	0.024	-0.10	0.921	0.024	-0.10	0.921	0.000	-1.75	0.08
23	0.024	-0.10	0.921	0.024	-0.10	0.921	0.000	-1.75	0.08
24	0.024	-0.10	0.921	0.024	-0.10	0.921	0.000	-1.75	0.08
25	0.024	-0.10	0.921	0.024	-0.10	0.921	0.000	-1.75	0.08
26	0.024	-0.10	0.921	0.024	-0.10	0.921	0.000	-1.75	0.08
27	0.024	-0.10	0.921	0.024	-0.10	0.921	0.000	-1.75	0.08
28	0.024	-0.10	0.921	0.024	-0.10	0.921	0.000	-1.75	0.08
29	0.024	-0.10	0.921	0.024	-0.10	0.921	0.000	-1.75	0.08
30	0.024	-0.10	0.921	0.024	-0.10	0.921	0.000	-1.75	0.08
31	0.024	-0.10	0.921	0.024	-0.10	0.921	0.000	-1.75	0.08
32	0.024	-0.10	0.921	0.024	-0.10	0.921	0.000	-1.75	0.08
33	0.024	-0.10	0.921	0.024	-0.10	0.921	0.000	-1.75	0.08

Table 11. Complete list with z-scores and variance for every negatively oversampled triple (WN18RR)

References

Akrami, F., Guo, L., Hu, W., & Li, C. (2018). Re-evaluating embedding-based knowledge graph completion methods. In *Proceedings of the 27th acm international conference on information and knowledge management* (pp. 1779–1782).

Ali, M., Hoyt, C. T., Domingo-Fernández, D., & Lehmann, J. (2019). Predicting missing links using pykeen. In *Iswc satellites* (pp. 245–248).

Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., & Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, *26*.

Brandenburg, F. J., Gleißner, A., & Hofmeier, A. (2013). Comparing and aggregating partial orders with kendall tau distances. *Discrete Mathematics, Algorithms and Applications*, *5*(02), 1360003.

Ji, S., Pan, S., Cambria, E., Marttinen, P., & Philip, S. Y. (2021). A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Transactions on Neural Networks and Learning Systems*.

Kazemi, S. M., & Poole, D. (2018). Simple embedding for link prediction in knowledge graphs. *Advances in neural information processing systems*, *31*.

Kolda, T. G., & Bader, B. W. (2009). Tensor decompositions and applications. *SIAM review*, *51*(3), 455–500.

Kong, X., Chen, X., & Hovy, E. (2019). Decompressing knowledge graph representations for link prediction. *arXiv preprint arXiv:1911.04053*.

Kontonasios, K.-N., Spyropoulou, E., & De Bie, T. (2012). Knowledge discovery interestingness measures based on unexpectedness. *Wiley interdisciplinary reviews: data mining and knowledge discovery*, *2*(5), 386–399.

Kotnis, B., & Nastase, V. (2017). Analysis of the impact of negative sampling on link prediction in knowledge graphs. *arXiv preprint arXiv:1708.06816*.

McGarry, K. (2005). A survey of interestingness measures for knowledge discovery. *The knowledge engineering review*, *20*(1), 39–61.

Nickel, M., Murphy, K., Tresp, V., & Gabrilovich, E. (2015). A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, *104*(1), 11–33.

Nickel, M., Tresp, V., & Kriegel, H.-P. (2011). A three-way model for collective learning on multi-relational data. In *Icml.*

Pujari, M., & Kanawati, R. (2012). Supervised rank aggregation approach for link prediction in complex networks. In *Proceedings of the 21st international conference on world wide web* (pp. 1189–1196).

Pusala, M. K., Benton, R. G., Raghavan, V. V., & Gottumukkala, R. N. (2017). Supervised approach to rank predicted links using interestingness measures. In *2017 ieee international conference on bioinformatics and biomedicine (bibm)* (pp. 1085–1092).

Silberschatz, A., & Tuzhilin, A. (1995). On subjective measures of interestingness in knowledge discovery. In *Kdd* (Vol. 95, pp. 275–281).

Szilagyi, A., Grimm, V., Arakaki, A. K., & Skolnick, J. (2005). Prediction of physical protein–protein interactions. *Physical biology*, *2*(2), S1.

Wang, M., Qiu, L., & Wang, X. (2021). A survey on knowledge graph embeddings for link prediction. *Symmetry*, *13*(3), 485.

Wang, Q., Mao, Z., Wang, B., & Guo, L. (2017). Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, *29*(12), 2724–2743.

Yang, Y., Lichtenwalter, R. N., & Chawla, N. V. (2015). Evaluating link prediction methods. *Knowledge and Information Systems*, *45*(3), 751–782.

Yang, Z., Ding, M., Zhou, C., Yang, H., Zhou, J., & Tang, J. (2020). Understanding negative sampling in graph representation learning. In *Proceedings of the 26th acm sigkdd international conference on knowledge discovery & data mining* (pp. 1666–1676).

Zhang, P., Qiu, D., Zeng, A., & Xiao, J. (2018). A comprehensive comparison of network similarities for link prediction and spurious link elimination. *Physica A: Statistical Mechanics and its Applications*, *500*, 97–105.