

# Detection of Relation Assertion Errors in Knowledge Graphs\*

André Melo

University of Mannheim  
B6 26, 68161 Mannheim, Germany  
andre@informatik.uni-mannheim.de

Heiko Paulheim

University of Mannheim  
B6 26, 68161 Mannheim, Germany  
heiko@informatik.uni-mannheim.de

## ABSTRACT

Although the link prediction problem, where missing relation assertions are predicted, has been widely researched, error detection did not receive as much attention. In this paper, we investigate the problem of error detection in relation assertions of knowledge graphs, and we propose an error detection method which relies on path and type features used by a classifier for every relation in the graph exploiting local feature selection. We perform an extensive evaluation on a variety of datasets, backed by a manual evaluation on DBpedia and NELL, and we propose and evaluate heuristics for the selection of relevant graph paths to be used as features in our method.

## CCS CONCEPTS

• Information systems → Data cleaning; • Computing methodologies → Semantic networks; Statistical relational learning; Machine learning;

## KEYWORDS

Knowledge graphs refinement, error detection, data cleansing

### ACM Reference format:

André Melo and Heiko Paulheim. 2017. Detection of Relation Assertion Errors in Knowledge Graphs. In *Proceedings of ACM Conference, Washington, DC, USA, July 2017 (Conference'17)*, 8 pages.  
<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

## 1 INTRODUCTION

Many of the knowledge graphs published as Linked Open Data have been created from semi-structured or unstructured sources. The magnitude of many of these knowledge graphs, e.g.: DBpedia, NELL, Wikidata, YAGO, do not allow for manual curation, and, instead, require the use of heuristics. Such heuristics, however, do not guarantee that the resulting graphs are free from errors. Wikipedia, which serves as source for DBpedia and YAGO, is estimated to have 2.8% of its statements wrong [36], which add up to the error caused by the extraction heuristics. Therefore, automatic approaches to automatically detect wrong statements are an important tool for the improvement of knowledge graph quality.

\*Detection of Relation Assertion Errors in Knowledge Graphs

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).  
*Conference'17, July 2017, Washington, DC, USA*  
© 2017 Copyright held by the owner/author(s).  
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.  
<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

Incompleteness is another major problem of most knowledge graphs. Automatic knowledge graph completion has been widely researched [19], with a variety of methods proposed, including embedding models. Although such methods can also be trivially employed for error detection, their performance has not yet been extensively evaluated on the task.

Many existing large-scale error detection methods rely exclusively on the types of subject and object of a relation [7, 25, 26], and try to spot violations of the underlying ontology and/or typical usage patterns. While types can be a valuable feature, some knowledge graphs lack this kind of information, have only incomplete type information, or have types which are not very informative. Moreover, some errors might contain wrong instances of correct types. For example, if someone adds the fact `playedFor(Ronaldo, Manchester_United)`, which would be wrong because Ronaldo refers to Ronaldo Nazário instead of Cristiano Ronaldo, such an approach would not be able to detect the error.

In knowledge graph completion, paths in the graph have been proven to be valuable features [8, 11]. For instance, in order to predict whether a person  $a$  lives in a place  $b$  (`livesIn(a,b)`), one important path feature is whether the person has a spouse who lives in  $b$  (`spouse(a,X) → livesIn(X,b)`), or whether the person has some child who was born in  $b$  (`childOf(X,a) → bornIn(X,b)`). Generalizing it for any pair of entities in a given relation, we can simply consider the previous example as path features `spouse → livesIn` and `childOf-1 → bornIn`, with binary values indicating if the entities pair can be connected through each of the paths. For error detection, these features can complement the type features. However, searching for interesting paths for all the relations in a knowledge graph can be a challenging task, especially in datasets with many relations.

In this paper, we propose a hybrid approach called *PaTyBRED* (Paths and Types with Binary Relevance for Error Detection), a method for the detection of relation assertion errors in knowledge graphs, which incorporates type and path features into local relation classifiers. Furthermore, we propose heuristic measures for the exploration of the paths search space. We perform an extensive comparison of our approach with state-of-the-art error detection and knowledge completion methods, and we conduct a manual evaluation of our approach on DBpedia and NELL.

## 2 PROBLEM DEFINITION

The problem addressed in this paper is the detection of erroneous relation assertions in knowledge graphs. A dataset containing errors is given, and the facts should be ranked by their likelihood of being wrong.

It is important to note that we consider only features which can be extracted from the links between entities (`owl:ObjectProperty` relation assertions) and types (`rdf:type` assertions). To make the

approach as versatile and applicable to as many knowledge graphs as possible, we do not use any other information, such as textual or numerical literals, or external knowledge sources. The problem can be defined as relation assertions error detection on internal features according to [23].

### 3 RELATED WORK

The problem of relation assertion error detection in knowledge graphs has been researched by the Semantic Web community. A few methods have been proposed for cleansing large-scale LOD knowledge graphs, such as DBpedia and NELL, which contain many relation assertion errors that cannot be detected by reasoning methods [25]. Absence of domain and range restrictions of relations or too general restrictions is one of the main causes of such problems.

SDValidate [25] exploits statistical distributions of types and relations, and [7] applies outlier detection on type-based entity similarity measures to detect erroneous relation assertions. These methods can effectively detect errors on DBpedia, however they require the existence of informative type assertions. Moreover, more complex errors containing wrong entities with correct types cannot be detected. A detailed survey including link prediction and error detection methods for knowledge graphs can be found in [23].

Knowledge graph completion (KGC) is a problem highly related to error detection. Despite being a different problem, KGC methods can also be used on the problem addressed in this paper. This kind of methods can be divided into graph-based, which relies on features which can be directly observed in the graph, and embedding methods, which learn latent features that represent entities and relations in an embedding space.

The Path Ranking Algorithm (PRA) [11] has shown that a logistic regression classifier using path features generated with random walks can be used for learning and inference in KGs and outperforms N-FOIL horn-clause inference on NELL [13]. The approach has been improved with Sub-graph Feature Extraction (SFE) [8], which also simplifies aspects of PRA. For instance, while PRA uses real-value features which correspond to the probabilities to reach  $o$  from  $s$  with a given path, SFE simply uses binary features which indicate if  $o$  can be reached from  $s$  or not. SFE not only reduces runtime by an order of magnitude when compared with PRA, but it also improves the qualitative performance.

In the recent years, knowledge graph embedding models, i.e., projections of knowledge graphs into lower-dimensional, dense vector spaces, have received a lot of attention [34]. Several different models have been developed for the knowledge graph completion problem and have brought improvements in performance.

There is a plethora of different embeddings models for knowledge graphs. One of the earliest embedding models is RESCAL [21], which performs tensor factorization on the knowledge graph's adjacency tensor, with the resulting eigenvectors corresponding to the entity embeddings and the core tensor the relations matrices. TRESICAL [5] extends RESCAL by exploiting entity types as well as domain and range restrictions of relations to improve the data quality and speed up the tensor factorization process. Neural Tensor Model (NTN) [30] represents each relation as a bilinear tensor operator followed by a linear matrix operator. Other early embedding

models include Structure Embeddings (SE) [3], Semantic Matching Energy (SME) [1] and Latent Factor Model (LFM) [9].

Translation-based embeddings represent relations as translations between subject and object entities. TransE [2] was the first translation-based model and entities and relations share the same embeddings space. In TransH [35] and TransR [15] the translations are performed in the relations space, which is different from the entities space, and require projection matrices to map the entities onto the relations space. TransG [37] and CTransR [15] incorporate multiple relation semantics, where a relation may have multiple meanings determined by the entities pair associated with the relation. PTransE [14] extends TransE by considering relation paths as regular relations, which makes the number of relations considered grow exponentially.

Other approaches include DistMult [38], which uses dot product instead of translations to compute the triple scores. HolE [20] used circular correlation as an operator to combine the subject and object embeddings, Complex Embeddings [32] represents a triple score as the hermitian dot product of the relation, subject and object embeddings, which consist of real and imaginary vector components. ProjE [29] formulates the knowledge graph completion as a ranking problem, and it optimizes the ranking of candidate entities collectively. It is reportedly the best performing KGC method. Some embedding models, such as RDF2Vec [28] and Global RDF vectors [18], are not conceived for the KGC task and cannot generate triple scores. Thus they cannot be directly used for error detection in the same way the other models mentioned earlier can.

Recently some works have raised doubts about the performance of new KGC embeddings models. Most of the experiments rely exclusively on two datasets (WN18 and FB15k), which contain many inverse relations citeobserved-versus-latent-features-for-knowledge-base-and-text-inference. Therefore some of the models may exploit this characteristic and not necessarily perform as well on other KGs. It has also been shown that the presence of relations between candidate pairs can be an extremely strong signal in some cases [31]. Moreover, recent works showed that a hyperparameter tuning has been overlooked and that a simple method, such as DistMult, can achieve state-of-the-art performance when well tuned [10].

### 4 APPROACH

Our proposed approach is inspired by the Path Ranking Algorithm (PRA) [11] and SDValidate [25]. It consists of a binary classifier for every relation which predicts the existence of a given pair of subject and object in the given relation. The set of classifiers can be thought of as a single multilabel classifier with binary relevance (i.e., each relation that can hold between a pair of instances is a label), where one binary classifier is learned for each class separately, and local feature selection [16], with different classifiers being able to work on different sets of specialized features.

We use two kinds of features. The first one are the types of subject and objects. This kind of information has been successfully used for error detection in SDValidate [25]. By analyzing the types of subject and object in one given relation, one can easily spot a very common kind of error without relying on the domain and range restrictions, which are often inexistent or too general. For example, in DBpedia the triple recordedIn(I'm\_a\_Loser,

Abbey\_Road) is wrong. I'm\_a\_Loser is a song by The Beatles from the album Abbey\_Road and the relation recordedIn has domain MusicalWork and range PopulatedPlace. A song being recorded in an Album is a clearly wrong fact. At the same time, if the object were Abbey\_Road\_Studio of the type Recording\_Studio, which is not a subclass of PopulatedPlace, the fact would also be wrong according to a method relying solely on types. If there are many facts where songs are recorded in recording studios, statistical methods such as SDValidate would be able to identify that such a pattern is common, and therefore unlikely to be wrong, despite the violation of range restriction, while a song recorded in album is uncommon, therefore likely to be an error. Hence, statistical approaches such as SDValidate respect the actual usage of the ontology, rather than its axiomatic design. Recent works have been proposed that pinpoint such mismatches automatically [22]. Moreover, type assertions might be absent or too general, resulting in no relevant information.

The main problem with this kind of approach is that it solely relies on type features. That means such approaches do not work on knowledge graphs with no type assertions, and may have poor performance on datasets with a shallow type hierarchy, with non-informative types, or with incomplete type assertions. Moreover, solely using type features, it is impossible to detect wrong facts with wrong entities of correct types, for instance, when a person instance is confused with another of same or similar name.

Alternatively we can use path features similar to those of PRA. However, solely relying on path features also has its problems. One of them is that correct facts may be labeled as error because of incompleteness. For instance, if river instances have the properties country (i.e., the countries a river passes through, typically multi-valued), and mouthCountry (i.e., the country where the river's mouth is, typically single-valued), then the feature country will be relevant for the relation mouthCountry since the confidence of the rule  $\text{mouthCountry}(X,Y) \Rightarrow \text{country}(X,Y)$  is close to 1. However, some rivers do not have any assertions for country because of incompleteness, thus their correct mouthCountry assertion is predicted to be wrong. That can lead to propagation of incompleteness.

Another problem is that since country is a more relevant feature to mouthCountry than vice versa, since the latter is far less common than the former. Hence, if an error occurs in the assertion of country for a river, it might happen that a correct mouthCountry assertion ends up being more likely to be detected as an error than the wrong country assertion. In order to make our approach more robust, we combine both type and path features.

Finding the relevant paths for each relation can be a challenging task. Since several paths may be relevant to different relations, we compute all possible paths up to a given length, and for every relation's local classifier we perform local feature selection. The number of possible paths grows exponentially with the number of relations, therefore an exhaustive search can easily become unfeasible. It is then crucial to have heuristics to efficiently navigate the search space. In the following subsection we propose and discuss such heuristic measures.

#### 4.1 Extracted Features

Our method includes the following parameters that define the path selection: maximum path length, maximum number of paths per level, and path selection heuristics. Following the approach described in [12], we use the domain and range restrictions of relations for pruning uninteresting paths, and we do not allow a relation to be immediately followed by its inverse. If the number of possible paths of a certain length exceeds the maximum number of paths per level, we apply our path selection heuristics to prune the least interesting paths and comply with the specified paths upper limit.

We define a knowledge graph  $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ , where  $\mathcal{T}$  is the T-box and  $\mathcal{A}$  is the A-box containing relations assertions  $\mathcal{A}_R$  and type assertions  $\mathcal{A}_C$ . We define  $N_C$  as the set of concepts (types),  $N_R$  as the set of relations and  $N_I$  as the set of individuals (entities which occur as subject or object in relations). The set of relation assertions is defined as  $\mathcal{A}_R = \{r(s, o) | r \in N_R \wedge s, o \in N_I\}$  and the set of type assertion as  $\mathcal{A}_C = \{C(s) | C \in N_C \wedge s \in N_I\}$ .

We define a path  $P$  as a sequence of relations  $r_1 \rightarrow \dots \rightarrow r_i \rightarrow \dots \rightarrow r_n$ . The sequence of relations is connected by a chain of variables, with  $P(s, o)$  meaning  $s$  and  $o$  can be connected by a path  $P(s, o) \iff r_1(s, x_1) \wedge \dots \wedge r_i(x_{i-1}, x_i) \wedge \dots \wedge r_n(x_{n-1}, o)$ . The inverse of a relation  $r$  is denoted as  $r^{-1}$  where  $r^{-1}(s, o) = r(o, s)$  can also be part of paths. A path of length one  $P = (r)$  is equivalent to the relation itself  $P(s, o) \equiv r(s, o)$ . The length of a path is denoted as  $|P|$ . We define the set of subjects of  $P$  as  $s_P = \{s | P(s, o)\}$  and set of objects as  $o_P = \{o | P(s, o)\}$ .

Relations and paths can be represented as adjacency matrices of size  $|N_I| \times |N_I|$ . The adjacency matrix of  $P$  can be computed by the dot product of its relations. However, computing the dot product of adjacency matrices can be an expensive operation, especially in large-scale knowledge graphs with millions of entities and high number of relations. Therefore, we need heuristic measures to explore the search space and compute the dot product only for the most relevant paths.

Let  $A$  and  $B$  be adjacency matrices – which can refer to a single relation or a path – which we want to concatenate in order to form a new path  $A \cdot B$ . We want a heuristic measure which can estimate the relevance of the path  $A \cdot B$  without having to perform a potentially expensive matrix multiplication to compute its adjacency matrix. Since the paths computed are to be used by all relations, the proposed heuristic measures should not be computed with respect to a target relation, but only consider the matrices  $A$  and  $B$ .

Paths with empty adjacency matrices ( $|A \cdot B| = 0$ ) are useless and should be pruned. A simple way to safely prune them is to calculate  $o_A \cap s_B$ . The set of objects  $o_A$  contains the columns of  $A$  which have non-zero elements, and the set of subjects  $s_B$  contains the rows of  $B$  which have non-zero elements. If the intersection is empty, then we know that  $|A \cdot B| = 0$ . Note that  $|s_B| \leq |B|$  and  $|o_A| \leq |A|$ , and the intersection is cheaper to compute than dot product, therefore the runtime for computing  $o_A \cap s_B$  is shorter.

For our proposed heuristic measures, we assume that paths with denser adjacency matrices are more likely to be more relevant features. Since the size of the intersection  $o_A \cap s_B$  can be a good indicator of the number of nonzero elements in  $A \cdot B$ , we use it to define three measures for estimating the relevance of a path

$A \times B$ : We employ that characteristic into three proposed relevance measures *inter*,  $m_1$  and  $m_2$  (c.f Equations 1, 2 and 3).

$$\text{inter}(A, B) = |o_A \cap s_B| \quad (1)$$

$$m_1(A, B) = \frac{|o_A \cap s_B|}{|s_A \cap o_B| + 1} \quad (2)$$

$$m_2(A, B) = |o_A \cap s_B| \times |s_A \cup o_B| \quad (3)$$

By early pruning irrelevant paths, time is saved not only by computing fewer adjacency matrices, but also the number of features to be considered is reduced (fewer columns in the features table to be populated and less features to have the relevance computed).

Once the relevant paths have been selected, we compute their adjacency matrices and use them to populate the features used to train the relation classifiers. One of the problems of computing the whole adjacency matrix of paths is that some can be very dense and require a lot of memory. For example, the path `birthPlace → locatedIn-1` on DBpedia, which represents everything which is located in a place where someone was born in. Its adjacency matrix contains around 100 million non-zero elements and consumes more than 1GB of memory. As it is unlikely that all the entries in the matrix will be used, it would be desirable to handle such cases in a more efficient manner in order to restrict the memory consumption and speed up the paths adjacency matrices computation process.

It is worth pointing that the `rdf:type` relation is not considered in the paths. They are treated separately and are used to generate the type features, which consist of the set of asserted and subsumed types of an instance (we materialize the subsumed types into the assertions and ignore the subsumption relations). Integrating types into the paths can be problematic. Firstly it would significantly increase the search space. Secondly, a path which begins with the `rdf:type`, can only continue with `rdf:type-1` because types can only be objects in this relation (if we do not consider OWL class axioms in paths), and as mentioned earlier, we do not allow a relation to be immediately followed by its inverse.

## 4.2 Learning the Model

Once the paths have been selected, and their adjacency matrices have been computed, we can use them together with types as features to predict the existence of an entity pair  $(s, o)$  in a relation. The first step is to build a training dataset containing all extracted features for each relation  $r$ . We use as positive examples the entity pairs  $D_{\text{pos}} = \{(s, o) | r(s, o)\}$ , i.e. all the non-zero cells in the relation's adjacency matrix. Following [2], we generate negative instances  $D_{\text{neg}} = \{\gamma(s, o) | (s, o) \in D_{\text{pos}} \wedge \gamma(s, o) \notin D_{\text{pos}}\}$  for supervised training by corrupting entity pairs with  $\gamma$ , which substitute the subject or the object for a random entity instance and ensuring the new pair is not positive. In a preliminary experiment, we compared this approach with that of [12], which is more expensive, and no significant difference in performance was observed.

As labels we use information from  $r$  indicating the existence of  $(s, o)$  in the relation. We extract path features from  $\mathcal{A}_R$  and type features from  $\mathcal{A}_C$ . The path features are boolean values indicating whether a path connects  $s$  to  $o$  ( $P(s, o) | \forall P \in \mathcal{P} - (r)$ ). The type features consist of the types of  $s$  and  $o$  (including subsumed types),

i.e.  $\{C|C(s)\}$  and  $\{C|C(o)\}$ . Other possible path feature is the existence of a path starting or ending in  $s$  and  $p$  ( $P(s, X)$ ,  $P(X, s)$ ,  $P(o, X)$ ,  $P(X, o)$ ) as proposed in SFE [8], however the authors found out that this kind of feature does not improve performance. Our experimental results confirmed their results, therefore we do not consider this kind of feature in our approach.

Before we learn the local classifiers, we evaluate the relevance of the features. Since different features might be relevant for different relations, we perform feature selection separately for every relation. This allows the relation classifiers to work on a small set of locally relevant features, and, at the same time, removes irrelevant features which might act as noise and reduce the classifier's performance [16]. We use the filter method, which simply select the top- $k$  most relevant features, with  $\chi^2$  as relevance measure.

When comparing PaTyBRED with PRA and SFE, our approach has the following advantages. We try different popular classifiers to learn the relations, and we found that logistic regression, which is used in PRA and SFE, is not the best performer. We introduce a local feature selection step prior to training the relation classifiers. We propose heuristic measures to explore the paths search space. Moreover, negative evidence features, i.e. paths which connect negative but no positive entity pairs of a relation, are also considered. Since our approach is supervised and includes negative examples in the training data, this kind of features are extremely important to identify wrong facts.

## 5 EXPERIMENTS

In our experiments, we evaluate the impact of different parameter settings in our approach, and compare it with SDValidate and state-of-the-art knowledge graph completion methods. We use ProjE<sup>1</sup> as well as the TransE and HolE implementations of scikit-kge<sup>2</sup>. The implementation of PaTyBRED is available on Github<sup>3</sup>. We do not directly compare our method with SFE, but we evaluate our approach with path features only (PaBRED), which perform at least as well as SFE.

The reported results from the embedding methods were obtained by not considering the type assertions. We tried adding the type assertions as an extra relation, however, this did not improve the results. The embedding methods suffer from the problem that the distribution of scores over different relations is not uniform. Often some relations have average triple scores lower than others, and this can result in a bias when detecting errors.

In order to reduce this problem, we run isolation forest to detect score outliers of each relation separately, and we use the outlier confidence values instead of the triple scores to rank the facts. Since unusually high scores are also outliers and we are interested only in the outliers of low scores, we do not consider as outlier any fact with score greater than the relation's average.

### 5.1 Datasets

In our experiments we use a variety of knowledge graphs, some of which are clean, and others noisy. In the first part of our experiments we automatically evaluate the performance of the error detection

<sup>1</sup><https://github.com/nddsg/ProjE>

<sup>2</sup><https://github.com/mnick/scikit-kge>

<sup>3</sup><https://github.com/aolimelo/kged>

algorithms. In order to make the evaluation automatic, we use a variety of datasets to which we add synthesized wrong facts. We generate the erroneous facts by corrupting the subject or object of true facts, i.e., replacing the original entity with a randomly selected which results in a fact which does not exist in the original data. For our generation process, we add 1% of noise, and we generate two kinds of errors. In the first, we corrupt the triple by selecting any of the entities from the knowledge graph (independent of type), and in the second, we select only triples which have the same types as the original entity. That means the errors of the second kind are, in principle, more difficult to be detected than those of the first kind, since the new entity is more likely to have characteristics similar to those of the original one.

The datasets used are the following: As input knowledge graphs, we use DBpedia (2015-10), and NELL (08m-690). We use the following smaller domain specific datasets: Semantic Bible<sup>4</sup> AIFB portal<sup>5</sup> and Nobel Prize<sup>6</sup>. We also select four of the largest conference datasets from the Semantic Web dog food corpus<sup>7</sup>, i.e., LREC2008, WWW2012, ISWC2013, and ESWC2015. WN18 and FB15k (WordNet 1.8 and a subset of Freebase with 15 000 entities), which have been widely used on link prediction experiments, are also used.

The Semantic Web dog food datasets are known to be correct and *locally* complete, i.e. no errors or missing relations *between contained entities*, therefore, the generated errors can be used as gold standard. We could not find any evaluation the of quality of AIFB, Semantic Bible or Nobel Prize. Since we cannot guarantee the quality of the data, the synthesized errors can be considered a silver standard. Because of incompleteness, some of the generated errors might actually be correct facts, meaning there can be false positives in the silver standard, and because of noise, there can also be false negatives in the silver standard.

The number of false positives is likely to be low even for highly incomplete datasets, since in general, the number of missing facts is significantly smaller than the number of possible facts ( $|N_R||N_I|^2 - |\mathcal{A}_R|$ ) from which the generated wrong facts are drawn.

In the second part of the experiments we use DBpedia and NELL as large-scale real-world use cases. These datasets are known to be noisy and incomplete, with type assertion completeness estimated to be at most 63.7% on DBpedia [25]. We do not synthesize any erroneous facts, and rank all the facts by their confidence values. Since we do not know the noisy facts or even the number of errors which exist in DBpedia, we manually evaluate the top-100 results.

## 5.2 Evaluation Measures

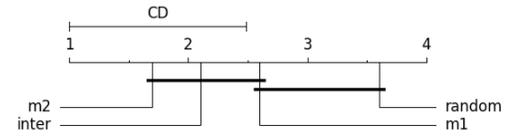
In our defined problem we use ranking measures to evaluate the performance of the error detection algorithms, since we compute scores for every triple in the graph and generate a ranking. Similar to link prediction papers we use the mean rank ( $\mu R$ ), mean reciprocal rank (MRR), as well as their filtered variations  $f\mu R$  and  $fMRR$  (c.f. Equations 4 and 5), which filters out correctly higher ranked predictions.

<sup>4</sup><http://www.semanticbible.com/>

<sup>5</sup>[http://www.aifb.kit.edu/web/Web\\_Science\\_und\\_Wissensmanagement/Portal](http://www.aifb.kit.edu/web/Web_Science_und_Wissensmanagement/Portal)

<sup>6</sup>[http://www.nobelprize.org/nobel\\_organizations/nobelmedia/nobelprize\\_org/developer/manual-linkeddata/terms.html](http://www.nobelprize.org/nobel_organizations/nobelmedia/nobelprize_org/developer/manual-linkeddata/terms.html)

<sup>7</sup><http://data.semanticweb.org/dumps/conferences/>



**Figure 1: Critical distance diagram comparing path selection heuristics**

$$fMRR = \frac{1}{|E|} \sum_{i=1}^{|E|} \frac{1}{rank_i - i + 1} \quad (4)$$

$$f\mu R = \frac{1}{|E|} \sum_{i=1}^{|E|} rank_i - i + 1 \quad (5)$$

We define  $E$  as the set of erroneous facts ordered by their rank in ascending order. Subtracting  $i - 1$  from the rank ensures that better ranked true positives are filtered out. For instance, if  $E = (1, 2, 3, 5, 8)$  its filtered sequence of ranks would be  $(1, 1, 1, 2, , 5)$ .

## 5.3 Parameter Settings

First, we evaluate how the different PaTyBRED parameters affect its performance. The evaluated parameters are the maximum path length ( $mpl$ ), the maximum number of paths per level ( $mppl$ ), the path selection heuristic measure ( $pshm$ ), the number of locally selected features ( $k$ ), and the local classifier ( $clf$ ).

As far as the maximum path length ( $mpl$ ) is concerned, the best results were achieved with  $mpl = 2$ , that is direct links and triangular patterns. Equivalent, inverse, and subproperty relations, as well as other kinds of associations can be exploited with direct links, while more complex associations with composed relations can be exploited with the triangular patterns. In none of the datasets used in our experiments, a  $mpl > 2$  achieved better results. It seems that paths longer than two do not bring any information gain, while it significantly increase the search space and slows runtime.

In our experiments, we evaluate three different classifiers ( $clf$ ): random forests (RF) [4], support vector machines (SVM) [6] and logistic regression (LR). We also try two different number of selected features  $k$ , i.e.,  $k = 10$  and  $k = 25$ . These numbers are low because we observed that only a small number of path and type features are relevant to the local relation classifiers. Table 1 show how the different settings of PaTyBRED<sub>25</sub><sup>SVM</sup> on various datasets. The results show that RF and SVM achieved the best results, while LR – which is used in PRA and SFE – lagged behind.

The heuristic measures used for selecting relevant adjacency matrices are those proposed in Section 4.1, i.e., *inter*, *m1* and *m2*. As a baseline, we use the *random* selection of paths. In order to better evaluate the quality of the paths selected we exclude the type features and consider exclusively the selected paths. We compared the heuristic measures on all the datasets presented in Section 5.1, ranked the measures and averaged them. In order to find out the significance of the results we perform Nemenyi Test with  $\alpha = 0.05$ . Since the number of datasets is rather small, the difference between *inter* and *m2* is not significant, however, they are significantly better than the random approach (c.f. Figure 1).

	$fMRR$										$f\mu R$								
	sembib	eswc	iswc	www	lrec	nobel	aifb	wn18	fb15k		sembib	eswc	iswc	www	lrec	nobel	aifb	wn18	fb15k
PaTyBRED <sup>LR</sup> <sub>10</sub>	0.800	0.835	0.811	0.212	0.754	0.690	0.014	0.584	0.618		0.008	0.020	0.006	0.0023	0.011	0.076	0.041	0.00352	0.015
PaTyBRED <sup>RF</sup> <sub>10</sub>	0.840	0.927	0.933	0.559	0.747	0.680	0.120	0.860	0.770		0.009	0.009	0.010	0.0003	0.006	0.080	0.031	0.00003	0.018
PaTyBRED <sup>SVM</sup> <sub>10</sub>	0.838	0.906	0.980	0.414	0.844	0.673	0.070	0.820	0.713		0.011	0.012	0.008	0.0007	0.004	0.103	0.041	0.00003	0.014
PaTyBRED <sup>LR</sup> <sub>25</sub>	0.745	0.907	0.862	0.707	0.786	0.788	0.068	0.584	0.524		0.005	0.022	0.003	0.0012	0.011	0.051	0.035	0.00349	0.014
PaTyBRED <sup>RF</sup> <sub>25</sub>	0.881	0.928	0.964	0.795	0.653	0.782	0.213	0.795	0.545		0.003	0.028	0.010	0.0001	0.006	0.051	0.028	0.00004	0.020
PaTyBRED <sup>SVM</sup> <sub>25</sub>	0.848	0.860	0.980	0.537	0.822	0.788	0.045	0.570	0.765		0.007	0.015	0.006	0.0003	0.005	0.063	0.028	0.00006	0.014

Table 1: Comparison of local classifiers and number of selected features on generated errors of kind 1

## 5.4 Comparison

Tables 2 and 3 report a comparison between PaTyBRED and the other state-of-the-art models. Table 3 refers to the datasets with errors with wrong entities of correct types and Table 2 refers to errors with wrong entities of any types. Table 3 does not contain results for WN18 and FB15k because the original datasets do not contain entity types, which prevents errors of kind 2 to be generated. For the same reason the results of SDValidate and TyBRED in Table 2 are not reported for WN18 and FB15k. We report values for  $fMRR$  and  $f\mu R$  ( $f\mu R$  values divided by the total number of facts in the KB in order to make the values more comparable).

It is noticeable that the results for AIFB are significantly worse than other datasets. One of the reasons is the fact that it has no inverse relations, which can be extremely helpful on the error detection. Another reason is the fact that in AIFB the author is defined by 27 `author_n` relations, with  $n$  indicating the position in the authors list. That means it is necessary to not only model the author relation, but also all the  $n^{\text{th}}$ -author relations.

PaTyBRED, TyBRED and PaBRED were run with 6 different configuration:  $clf \in \{LR, RF, SVM\}$  and  $k \in \{10, 25\}$ . For each dataset the results of the best performing configuration are reported. The maximum number of paths per level is set to 1000 and `m2` is used as heuristic measure when the number of possible paths exceeds 1000, and the maximum path length is set to 2. The values reported for the embeddings methods were the best amongst number dimensions  $d \in \{5, 15, 50, 100, 200\}$  and with the outlier detection, as explained earlier.

It is worth mentioning that the outlier detection helped improve the performance of embeddings'  $f\mu R$  performance on average on 15%. The best results for the embedding methods were obtained with  $d = 15$  or  $d = 50$  depending on the dataset. The results reported for the knowledge graph completion in the original paper for ProjE on FB15k were with  $d = 200$ . On error detection with the same dataset the best performance was with  $d = 50$ , cutting the  $f\mu R$  in half. Additionally,  $d = 5$  and  $d15$  also had better performance than  $d = 200$ . This indicates that when using embeddings for error detection, the dimensionality should be lower than for KGC. Since the dataset contains wrong triples, which shouldn't be fit by the model, overfitting can severely affect the performance (more than underfitting).

Our proposed method outperforms all the other methods, with the embedding methods having a surprisingly low performance. PaTyBRED performs best when combining types and paths, with

TyBRED (with types only) and PaBRED (with paths only) being generally worse. To further understand the importance of combining path type features, we analyze what kind of features are selected on the local classifiers and report the proportion of types and paths. Table 4 shows the average proportion of selected features over all relation classifiers with  $k = 10$ . Overall more type features are selected, but both kinds of features are relevant on the evaluated datasets. WN18 and FB15k are absent because they do not have type assertions, and therefore have only path features.

Table 3, where the erroneous facts contain wrong instances of correct types, shows how the performance of methods which rely on types exclusively (SDValidate and TyBRED) is similar to that of random ranking with  $f\mu R$  around 0.5. It also shows how detecting errors of kind 2 is more difficult than those of kind 1, and it reveals the importance of using path features for detecting facts with wrong instances of correct types. We can also observe that PaBRED has performance similar to PaTyBRED and even better on some datasets for kind 2 errors, since type features are useless to detect those errors, and not considering type features ensures that these cannot potentially replace more useful path features. The only exceptions are on LREC and AIFBportal, where PaTyBRED has better  $fMRR$  than PaBRED. However, on the same datasets PaBRED performs better in terms of  $f\mu R$ , meaning that it has better average rank but less highly ranked instances.

## 5.5 Manual Evaluation

In this section we perform a manual evaluation of PaTyBRED on two large-scale noisy datasets: DBpedia and NELL. We have a deeper look at the top-100 results and classify the triples as correct, wrong and *other errors*, i.e., correct triples with related errors, e.g. wrong or missing types of subject or object.

The results are shown in Figure 2 with PaTyBRED<sup>RF</sup><sub>10</sub> and PaTyBRED<sup>RF</sup><sub>25</sub> on DBpedia (dbp10, dbp25) and NELL (nell10, nell25). PaTyBRED seems to perform better on DBpedia with less local features (10) and more on NELL (25). Most of the other error cases occurred because of type assertion incompleteness, with the subject or object often having no types at all. Deleting these triples would lead to propagation of incompleteness. These cases could be automatically detected, and some of them fixed if the type completion methods [17, 24] are combined with error detection. The quality of predicted types can be asserted by the improvement of the scores of triples containing the entities with predicted types.

Some of the errors come from mistakes when linking Wikipedia pages with very similar names. One example of such problem is the

	$fMRR$									$f\mu R$								
	sembib	eswc	iswc	www	lrec	nobel	aifb	wn18	fb15k	sembib	eswc	iswc	www	lrec	nobel	aifb	wn18	fb15k
PaTyBRED	0.848	0.928	0.980	0.795	0.844	0.788	0.213	0.860	0.770	0.003	0.009	0.003	0.0001	0.004	0.051	0.028	0.00003	0.014
TyBRED	0.463	0.782	0.315	0.744	0.693	0.758	0.205	—	—	0.121	0.083	0.102	0.0740	0.113	0.084	0.085	—	—
PaBRED	0.800	0.831	0.980	0.503	0.778	0.200	0.173	0.860	0.770	0.009	0.010	0.005	0.0008	0.004	0.227	0.056	0.00003	0.014
SDValidate	0.265	0.140	0.218	0.109	0.307	0.464	0.022	—	—	0.355	0.397	0.326	0.3768	0.339	0.286	0.293	—	—
ProjE	0.102	0.175	0.047	0.098	0.138	0.187	0.048	0.004	0.014	0.149	0.197	0.201	0.1796	0.179	0.177	0.252	0.18714	0.125
HolE	0.011	0.018	0.025	0.018	0.065	0.026	0.001	0.002	0.006	0.204	0.258	0.108	0.1170	0.108	0.213	0.235	0.17304	0.083
TransE	0.058	0.001	0.000	0.001	0.039	0.051	0.005	0.001	0.000	0.226	0.302	0.280	0.2381	0.163	0.320	0.329	0.26174	0.190

Table 2: Comparison of FMRR on generated errors of kind 1

	$fMRR$								$f\mu R$							
	sembib	eswc	iswc	www	lrec	nobel	aifb		sembib	eswc	iswc	www	lrec	nobel	aifb	
PaTyBRED	0.482	0.553	0.941	0.609	0.532	0.022	0.272	0.082	0.124	0.023	0.035	0.027	0.250	0.080		
TyBRED	0.001	0.001	0.001	0.001	0.000	0.000	0.000	0.597	0.503	0.512	0.495	0.551	0.526	0.496		
PaBRED	0.579	0.567	0.941	0.625	0.486	0.250	0.205	0.086	0.099	0.017	0.023	0.011	0.212	0.065		
SDValidate	0.001	0.001	0.001	0.000	0.000	0.000	0.000	0.570	0.457	0.467	0.506	0.495	0.495	0.475		
ProjE	0.064	0.026	0.015	0.026	0.007	0.067	0.018	0.215	0.362	0.223	0.245	0.254	0.274	0.269		
HolE	0.022	0.015	0.043	0.049	0.059	0.053	0.004	0.240	0.324	0.192	0.190	0.192	0.294	0.246		
TransE	0.092	0.004	0.012	0.000	0.012	0.001	0.003	0.247	0.308	0.239	0.337	0.148	0.413	0.339		

Table 3: Comparison of FMRR on generated errors of kind 2

	sembib	eswc	iswc	www	lrec	nobel	aifb	nell	dbpedia
Paths	0.432	0.412	0.415	0.358	0.479	0.222	0.182	0.032	0.060
Types	0.568	0.588	0.585	0.642	0.521	0.778	0.818	0.968	0.940

Table 4: Proportion of path and type features selected

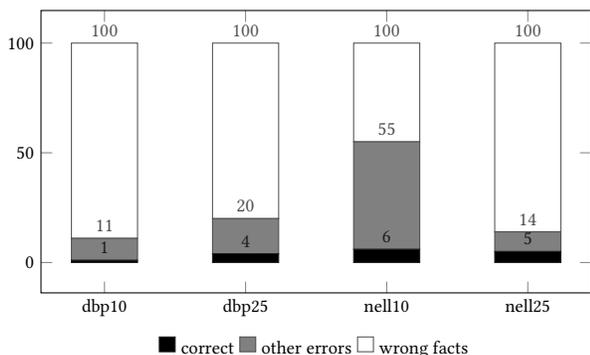


Figure 2: Manual evaluation on DBpedia and NELL

fact `formerTeam(Alan_Ricard, Buffalo_Bill)`, where the correct entity should be the NFL team `Buffalo_Bills` instead of the character `Buffalo_Bill`. An automatic approach which makes use of disambiguation links (in DBpedia the `dbo:wikiPageDisambiguates` relation) and string distance could in principle correct these errors. By replacing subject or object with their respective candidates and computing the triple scores, we can substitute the wrong triple with the best scoring candidate (a similar idea has been used for correcting links in Wikipedia [33]). In the manual evaluation, five of the DBpedia errors could potentially be fixed with such an approach.

Entities in DBpedia are described in much more detail than in NELL [27]. Around 20% of NELL’s instances are untyped, while in DBpedia only 1% of them have no types other than `owl:Thing`. Furthermore, in NELL, reasoning is already used in the construction process for error detection, which means that very obvious errors and violations of the underlying ontology are already removed. This may explain why NELL performs better with more locally selected features, as opposed to DBpedia. By increasing the number of features the number of correct facts with untyped subject or object in the top-100 was reduced from 48 to 9, and the number of actual errors increased from 45 to 86.

Amongst the five correct facts from DBpedia which were wrongly predicted to be errors, two were from the relation `seeAlso`. That is understandable since the relation has very wide semantics, and any pair of vaguely related entities can be correct. Modelling such a complex relation can be a difficult task. Another error detected was `location(Alan_Turing_Institute, British_Library)`, which is a correct fact, but the unique case of an organization which is located in a library. The last case is with the `foundedBy` relation, with two cases of newspapers found by political parties, not persons.

## 6 CONCLUSION

We have shown that although the error detection problem is similar to knowledge completion, methods which perform well in knowledge completion might not necessarily be appropriate for error detection. We propose PaTyBRED, a robust supervised error detection method which relies on type and path features, and compare it with state-of-the-art error detection and knowledge graph completion methods. We demonstrate the importance of combining those path and type features together, and we also perform a manual evaluation of our approach on DBpedia and NELL.

In the future, we plan to investigate the automatic correction of erroneous relation assertions by exploiting disambiguation links and entities string similarities in combination with error detection methods.

## ACKNOWLEDGMENTS

The work presented in this paper has been partly supported by the Ministry of Science, Research and the Arts Baden-Württemberg in the project SyKo<sup>2</sup>W<sup>2</sup> (Synthesis of Completion and Correction of Knowledge Graphs on the Web).

## REFERENCES

- [1] Antoine Bordes, Xavier Glorot, Jason Weston, and Yoshua Bengio. 2012. Joint Learning of Words and Meaning Representations for Open-Text Semantic Parsing. In *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2012, La Palma, Canary Islands, April 21-23, 2012*. 127–135. <http://jmlr.csail.mit.edu/proceedings/papers/v22/bordes12.html>
- [2] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. [n. d.]. Translating Embeddings for Modeling Multi-relational Data. In *Advances in Neural Information Processing Systems 26*.
- [3] Antoine Bordes, Jason Weston, Ronan Collobert, and Yoshua Bengio. 2011. Learning Structured Embeddings of Knowledge Bases. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2011, San Francisco, California, USA, August 7-11, 2011*. <http://www.aaai.org/ocs/index.php/AAAI/AAAI11/paper/view/3659>
- [4] Leo Breiman. 2001. Random Forests. *Machine Learning* 45, 1 (01 Oct 2001), 5–32. <https://doi.org/10.1023/A:1010933404324>
- [5] Kai-Wei Chang, Scott Wen-tau Yih, Bishan Yang, and Chris Meek. 2014. Typed Tensor Decomposition of Knowledge Bases for Relation Extraction. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*. <https://www.microsoft.com/en-us/research/publication/typed-tensor-decomposition-of-knowledge-bases-for-relation-extraction/>
- [6] Corinna Cortes and Vladimir Vapnik. 1995. Support-Vector Networks. *Mach. Learn.* 20, 3 (Sept. 1995), 273–297. <https://doi.org/10.1023/A:1022627411411>
- [7] Jeremy Debatista, Christoph Lange, and Sören Auer. 2016. A Preliminary Investigation Towards Improving Linked Data Quality Using Distance-Based Outlier Detection. In *Semantic Technology - 6th Joint International Conference, JIST 2016, Singapore, Singapore, November 2-4, 2016, Revised Selected Papers*. 116–124. [https://doi.org/10.1007/978-3-319-50112-3\\_9](https://doi.org/10.1007/978-3-319-50112-3_9)
- [8] Matt Gardner and Tom M. Mitchell. 2015. Efficient and Expressive Knowledge Base Completion Using Subgraph Feature Extraction. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*. 1488–1498. <http://aclweb.org/anthology/D/D15/D15-1173.pdf>
- [9] Rodolphe Jenatton, Nicolas L. Roux, Antoine Bordes, and Guillaume R Obozinski. [n. d.]. A latent factor model for highly multi-relational data. In *Advances in Neural Information Processing Systems 25*.
- [10] Rudolf Kadlec, Ondrej Bajgar, and Jan Kleindienst. 2017. Knowledge Base Completion: Baselines Strike Back. *CoRR* abs/1705.10744 (2017). <http://arxiv.org/abs/1705.10744>
- [11] Ni Lao and William W. Cohen. 2010. Relational Retrieval Using a Combination of Path-constrained Random Walks. *Mach. Learn.* 81, 1 (Oct. 2010), 53–67. <https://doi.org/10.1007/s10994-010-5205-8>
- [12] Ni Lao and William W. Cohen. 2010. Relational retrieval using a combination of path-constrained random walks. *Machine Learning* 81, 1 (01 Oct 2010), 53–67. <https://doi.org/10.1007/s10994-010-5205-8>
- [13] Ni Lao, Tom Mitchell, and William W. Cohen. 2011. Random Walk Inference and Learning in a Large Scale Knowledge Base. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP '11)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 529–539. <http://dl.acm.org/citation.cfm?id=2145432.2145494>
- [14] Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2015. Modeling Relation Paths for Representation Learning of Knowledge Bases. *CoRR* abs/1506.00379 (2015). <http://arxiv.org/abs/1506.00379>
- [15] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning Entity and Relation Embeddings for Knowledge Graph Completion. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI'15)*. AAAI Press, 2181–2187. <http://dl.acm.org/citation.cfm?id=2886521.2886624>
- [16] André Melo and Heiko Paulheim. 2017. Local and global feature selection for multilabel classification with binary relevance. *Artificial Intelligence Review* (2017), 1–28. <https://doi.org/10.1007/s10462-017-9556-4>
- [17] André Melo, Heiko Paulheim, and Johanna Völker. 2016. Type Prediction in RDF Knowledge Bases Using Hierarchical Multilabel Classification. In *Proceedings of the 6th International Conference on Web Intelligence, Mining and Semantics (WIMS '16)*. ACM, New York, NY, USA, Article 14, 10 pages. <https://doi.org/10.1145/2912845.2912861>
- [18] Simone Paolo Ponzetto Heiko Paulheim Michael Cochez, Petar Ristoski. 2017. Global RDF Vector Space Embeddings. In *International Semantic Web Conference*. to appear.
- [19] Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. 2016. A Review of Relational Machine Learning for Knowledge Graphs. *Proc. IEEE* 104, 1 (2016), 11–33. <https://doi.org/10.1109/JPROC.2015.2483592>
- [20] Maximilian Nickel, Lorenzo Rosasco, and Tomaso A. Poggio. 2015. Holographic Embeddings of Knowledge Graphs. *CoRR* abs/1510.04935 (2015). <http://arxiv.org/abs/1510.04935>
- [21] Maximilian Nickel, Volker Tresp, and Hans peter Kriegel. 2011. A Three-Way Model for Collective Learning on Multi-Relational Data. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*. ACM. [http://www.icml-2011.org/papers/438\\_icmlpaper.pdf](http://www.icml-2011.org/papers/438_icmlpaper.pdf)
- [22] Heiko Paulheim. 2017. Data-driven joint debugging of the DBpedia mappings and ontology. In *European Semantic Web Conference*. Springer, 404–418.
- [23] Heiko Paulheim. 2017. Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic Web* 8, 3 (2017), 489–508. <https://doi.org/10.3233/SW-160218>
- [24] Heiko Paulheim and Christian Bizer. [n. d.]. .
- [25] Heiko Paulheim and Christian Bizer. 2014. Improving the Quality of Linked Data Using Statistical Distributions. *Int. J. Semant. Web Inf. Syst.* 10, 2 (April 2014), 63–86. <https://doi.org/10.4018/ijswis.2014040104>
- [26] Heiko Paulheim and Aldo Gangemi. 2015. Serving DBpedia with DOLCE—more than just adding a cherry on top. In *International Semantic Web Conference*. Springer, 180–196.
- [27] Daniel Ringler and Heiko Paulheim. 2017. One Knowledge Graph to Rule them All? Analyzing the Differences between DBpedia, YAGO, Wikidata & co.. In *40th German Conference on Artificial Intelligence*. to appear.
- [28] Petar Ristoski and Heiko Paulheim. [n. d.]. .
- [29] Baoxu Shi and Tim Wening. 2017. ProjE: Embedding Projection for Knowledge Graph Completion. (2017). <https://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14279>
- [30] Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. 2013. Reasoning With Neural Tensor Networks for Knowledge Base Completion. In *Advances in Neural Information Processing Systems 26*. Curran Associates, Inc., 926–934. <http://papers.nips.cc/paper/5028-reasoning-with-neural-tensor-networks-for-knowledge-base-completion.pdf>
- [31] Kristina Toutanova and Danqi Chen. 2015. Observed Versus Latent Features for Knowledge Base and Text Inference. In 3rd Workshop on Continuous Vector Space Models and Their Compositionality. <https://www.microsoft.com/en-us/research/publication/observed-versus-latent-features-for-knowledge-base-and-text-inference/>
- [32] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Eric Gaussier, and Guillaume Bouchard. 2016. Complex Embeddings for Simple Link Prediction. *CoRR* abs/1606.06357 (2016). <http://arxiv.org/abs/1606.06357>
- [33] Chengyu Wang, Rong Zhang, Xiaofeng He, and Aoying Zhou. 2016. Error Link Detection and Correction in Wikipedia. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management (CIKM '16)*. ACM, New York, NY, USA, 307–316. <https://doi.org/10.1145/2983323.2983705>
- [34] Q. Wang, Z. Mao, B. Wang, and L. Guo. 2017. Knowledge Graph Embedding: A Survey of Approaches and Applications. *IEEE Transactions on Knowledge and Data Engineering* PP, 99 (2017), 1–1. <https://doi.org/10.1109/TKDE.2017.2754499>
- [35] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. [n. d.]. In *AAAI*.
- [36] Gabriel Weaver, Barbara Strickland, and Gregory Crane. 2006. Quantifying the accuracy of relational statements in Wikipedia: a methodology. *2006 IEEE/ACM 6th Joint Conference on Digital Libraries* 00 (2006), 358. <https://doi.org/doi.ieeeecomputersociety.org/10.1145/1141753.1141853>
- [37] Han Xiao, Minlie Huang, Yu Hao, and Xiaoyan Zhu. 2015. TransG : A Generative Mixture Model for Knowledge Graph Embedding. *CoRR* abs/1509.05488 (2015). <http://arxiv.org/abs/1509.05488>
- [38] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2014. Learning Multi-Relational Semantics Using Neural-Embedding Models. *CoRR* abs/1411.4072 (2014). <http://arxiv.org/abs/1411.4072>