Knowledge Graph Consolidation by Unifying Synonymous Relationships

Jan-Christoph Kalo(✉), Philipp Ehler, and Wolf-Tilo Balke

Institut für Informationssysteme, Technische Universität Braunschweig,
Mühlenpfordtstraße 23, 38106 Brunswick, Germany
{kalo,balke}@ifis.cs.tu-bs.de, p.ehler@tu-bs.de

Abstract. Entity-centric information resources in the form of huge RDF knowledge graphs have become an important part of today's information systems. But while the integration of independent sources promises rich information, their inherent heterogeneity also poses threats to the overall usefulness. To some degree challenges of heterogeneity have been addressed by creating underlying ontological structures. Yet, our analysis shows that synonymous relationships are still prevalent in current knowledge graphs. In this paper we compare state-of-the-art relational learning techniques to analyze the semantics of relationships for unifying synonymous relationships. By embedding relationships into latent feature models, we are able to identify relationships showing the same semantics in a data-driven fashion. The resulting relationship synonyms can be used for knowledge graph consolidation. We evaluate our technique on Wikidata, Freebase and DBpedia: we identify hundreds of existing relationship duplicates with very high precision, outperforming the current state-of-the-art method.

Keywords: Data quality · Synonym detection · Knowledge embedding

1 Introduction

Knowledge graphs (KG) efficiently collect entity-centric data in triple format and serve an increasing number of applications. Beginning with the Semantic Web standard RDF for knowledge representation, projects like Wikidata [27], DBpedia [4], Freebase [5], YAGO [25] and the Google Knowledge Vault [7] over the last years have grown significantly to support for instance Web search, question answering, and recommender systems.

But from the beginning, highly heterogeneous data items have caused severe problems in RDF databases, because a huge number of independent data sources needs to be integrated in a world-wide Semantic Web. During integration, heterogeneity issues are mostly manifested by having different RDF identifiers for the same real-world objects or relationships. However, while ontology alignment is extensively investigated for example at the Ontology Alignment Evaluation

C. Ghidini et al. (Eds.): ISWC 2019, LNCS 11778, pp. 276–292, 2019.
https://doi.org/10.1007/978-3-030-30793-6_16

Initiative[1] at ISWC, research has mainly focused on ontology and class alignments for two ontologies, often even requiring a complete OWL ontology. A detailed analysis of the DBpedia KG reveals that we are indeed facing another big problem: duplicates within the same KG. For instance, more than 26 different identifiers represent the birthplace relationship. For the entity Albert Einstein `dbo:birthPlace` is used, birthplaces of other persons use `dbp:birthCity` or even an identifier inspired by the French language `dbp:lieuDeNaissance`. Thus, queries asking for birthplaces using the `dbo:birthPlace` URI will be incomplete: persons whose birthplace is stated in some synonymous relationship will not be returned.

The problem of finding these *synonymous relationships* has hardly gotten any attention. Existing work [2] on this topic has only been evaluated on a small dataset, not reflecting the heterogeneities of today's large KGs. Traditional ontology alignment techniques often require two distinct ontologies as an input and are also pushed to their limits due to the lack of OWL statements in common KGs. Also natural language processing-based techniques like DOME from the OAEI 2018 [9] are often pushed to their limits here, because several KGs like Wikidata or Freebase use complex identifiers for naming relationships so that natural language techniques cannot be used.

In this paper, we detect synonymous relationships in a data-driven fashion only relying on the KG itself, thus not making any assumptions on the data: We are independent of a formal ontology in OWL and work with arbitrary identifiers for relationships. Our technique transfers ideas from synonym detection with word embeddings in natural language processing [17,22,30] into the field of KGs. Recently, relational learning techniques, also known as *knowledge embeddings*, have already been proposed to predict new triples in KGs [18,21,28]. In a nutshell, they are machine learning models trained on large sets of triples, learning latent vector representations of entities and relationships, which may be used to predict the correctness of known and unknown triples. We are the first work that makes use of the relationship representation in knowledge embeddings by showing that it may be used to reliably measure semantic similarity of knowledge graph's relationships. The main contributions of our work are:

- We develop a new method for identifying synonymous relationships in knowledge graphs by employing knowledge embeddings.
- Our method is purely data-driven not making any assumptions on the data and therefore is generalizable to all kinds of KGs.
- In an extensive evaluation with state-of-the-art knowledge embeddings (RESCAL [20], TransE [6], TransH [29], TransD [12], ComlEx [26], DistMult [31], HolE [19] and ANALOGY [16]) on Freebase, Wikidata and DBpedia, we demonstrate that we are able to identify synonyms with very high precision, outperforming a current state-of-the-art method.
- For reproducibility, we provide all our source code, datasets, and results in a publicly available Github repository.[2]

[1] http://oaei.ontologymatching.org/.

[2] https://github.com/JanKalo/RelAlign.

2 Related Work

Synonym Detection for relationships is about finding relationships with identical semantics within a single KG. To the best of our knowledge, only a single work on synonymous relationships [2] exists. Abedjan et al. [2] have noticed that particularly in DBpedia several synonymous predicates exist. To overcome problems in querying, they propose a query expansion process that builds on top of *synonymously used* relationships. They argue that for example the relationships `artist` and `starring`, even though they are not directly synonymous, in context of movies are synonymously used, making them good candidates for query expansion. Our manual analysis shows that the definition of synonymously used predicates is rather vague and differ from one application to another. Synonymous relationships as used in this paper are a subclass of synonymously used relationships, so the technique can serve as a baseline for this work.

The method of Abedjan et al. works with frequent item set mining. First, relationships that often co-occur for the same object entities are gathered in frequent item sets. Frequent item sets that exceed a certain minimum support threshold are further analyzed. The minimum support is an input parameter defined by the user, highly influencing precision and recall. All predicates within the same frequent item set are evaluated pairwise with the Reversed Correlation Coefficient together with their co-occurrence with the same subject entities. This is based on the assumption in mind that synonymous relationships should not co-occur for the same subject entities. In contrast, knowledge embedding based methods as proposed by us do not make any assumptions on the data. The authors evaluate their approach on a small manually built synonym dataset from DBpedia 3.7, Magnatune and Govwild. They show that their approach often achieves a precision value above 50%.

Ontology Alignment in contrast to synonym detection, is concerned with matching schemas of more than a single knowledge graph or RDF dataset. It has been a hot topic since the early days of the Semantic Web. Every year the Ontology Alignment Evaluation Initiative (OAEI) organizes a workshop for benchmarking different alignment systems. Its goal is to overcome problems like duplicate entities, classes and also relationships to integrate two or more ontologies [3,11,13,24]. Typically three different matching problems are addressed in the field of ontology alignment: Instance matching, class matching and sometimes also relationship matching. Instance matching or entity matching which is about finding synonymous entities between two or more knowledge bases [10,14]. These techniques rely on matching entities with similar relationships and properties. Class matching is about finding classes with equivalent semantics, relationship alignment about finding equivalent relationships.

DOME by Hertling et al. [9] is the only system that creates a relationship alignment in the knowledge graph track of OAEI 2018. However, its matching component relies on string similarity techniques, being very restrictive. Knowledge graphs often have complex identifiers as relationship URIs, making it impossible for such natural language based techniques to work at all.

Other ontology alignment tools (e.g. PARIS [24]) usually rely on two distinct ontologies and are not able to identify synonyms within a single knowledge graph, because their matching mechanism works on the relationships extensions, i.e. the entities taking part in the relations. In case of synonyms within a single knowledge graph this is usually not applicable, since synonymous relationships might have no overlap in their extension.

Furthermore, several ontology alignment systems that have been presented at OAEI over the last years are relying on a manually built ontology in OWL. They are not working on knowledge graphs that do not provide OWL information, as for example Wikidata and Freebase.

Knowledge Embeddings are usually used for predicting new triples in KGs, but can also be used for instance matching or entity resolution [18,28], which has some similarity to finding synonymous relationships. To the best of our knowledge there is only very few works that have looked concretely at the problem of finding instances of the same real-world entity with the help of knowledge embeddings. It has been proposed to formulate entity resolution as a link prediction task by predicting triples of the form (x, `owl:sameAs`, y) [20]. For RESCAL, Nickel et al. describe how to directly compare the entity representations to find identical entities, but they evaluate this idea only on a small dataset with about 2500 entities and only 7 relationships [20]. The idea of using relationship representations of knowledge embeddings has not been tested and evaluated before.

3 Preliminaries

In the Semantic Web, knowledge graphs are represented by the Resource Description Framework (RDF), a standard for knowledge representation by the W3C [1]. Knowledge in RDF has the form of subject, predicate, object *triples*: $(s, p, o) \in E \times R \times (E \cup L)$. Subjects stem from a set of resources E, representing entities or concepts (often from the real-world). Predicates stem from a set of relationships R. And objects are either resources like subjects or literals from the set L. They may be strings, numbers or dates. Resources and relationships are represented by Uniform Resource Identifiers (URIs). Due to better readability, in all our examples, we use textual labels instead of URIs for the identification of resources and relationships. Note that we focus on RDF without blank nodes and reification, since they cannot be processed by any of the knowledge embedding techniques we employ in this paper. A *knowledge graph* is a finite set of triples $KG \subseteq E \times R \times (E \cup L)$.

Since large KGs are usually created by crowd workers or automatic extraction, it may contain synonymous relationships or entities. With *synonymous* we refer to two (or more) distinct URIs either in E or R that refer to the same real-world entity, concept or relationship. As an example from DBpedia, the relationships `birthPlace` $\in R$ and `placeOfBirth` $\in R$ both refer to the relationship connecting a living being to its place of birth, which usually is a city. Similar to the work in [2], we are interested in finding *synonymous relationships* within a single knowledge base.

Given some knowledge graph *Knowledge Graph Consolidation* is the problem of finding all possible synonymous relationships so that they may be integrated. In the Semantic Web, these relationships are either collapsed into a single relationship or marked as identical by introducing a new triple with the `owl:sameAs` predicate.

4 Detecting Synonymous Relationships with Knowledge Embeddings

In this section, we present a new classification method for finding synonymous relationships in large KGs based on knowledge embedding techniques. Our idea is inspired by synonym search from natural language processing, which is often based on latent vector representations of words [17,22]. High-dimensional vector representations of RDF-based KGs (*knowledge embeddings*) are based on statistical relational learning techniques. For a detailed overview of existing knowledge embedding models is provided in the survey by Nickel et al. [18]. The latent vector representations of a knowledge embedding are learned from a KG by computing an optimization function on the set of correct triples from a KG and a set of automatically generated incorrect triples. During this optimization process, knowledge from the KG is encoded into an entity and a relationship representations which usually is combined to predict new triples. Empirical evaluations have shown that known triples from KG, but also unknown triples that have not been present in the KG are predicted by these models with high precision, at least when evaluated on small datasets like FB15K from Freebase and WN18 from Wordnet [6]. For entities it has been shown that their embeddings may be used to measure semantic similarity by applying distance metrics on the vectors [20,21].

Our approach uses a property of knowledge embeddings that has not been exploited before. We show that not only the entity representation can be used to measure semantic similarity, but also the latent representation of relationships can be used to measure its semantic similarity. Our work investigates the advantages and limits of this property for detecting synonymous relationships with knowledge embeddings, so relationships that have a very high semantic similarity. In this paper, we employ the knowledge embeddings RESCAL [20], TransE [6], TransH [29], TransD [12], ComplEx [26] DistMult [31], HolE [19] and ANALOGY [16]. From all models, we can obtain a relationship representation either in form of a vector, as a matrix, or as a concatenation of several matrices that can be used to measure the semantic similarity of the relationships in a vector space using classical vector metrics. Since knowledge embeddings are currently not able to embed literal values or relationships that are in triples with literal values, our method is restricted to relationships between resources.

4.1 Representing Relationships in a Knowledge Embedding

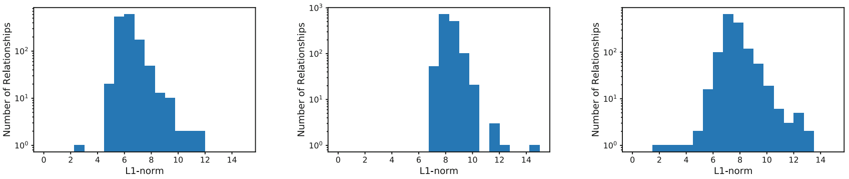
As already mentioned, knowledge embeddings have been created to predict new triples, usually by applying vector operations on subject, predicate, object vector representations. To give an intuition of why the techniques are suitable for finding synonymous relationships, we provide a small example: Given two true triples, (Albert_Einstein, birthplace, Ulm) and its synonymous counterpart (Albert_Einstein, bornIn, Ulm). Albert_Einstein and Ulm having unique vector representations in the knowledge embedding. The vector representation Albert_Einstein and the vector for birthplace can be combined in such a way that the vector of Ulm is predicted, using the prediction capabilities of the embedding. Since the same mechanism also works when combining the vector of Albert_Einstein and bornIn, usually the relationship vectors for birthplace and bornIn are identical. But also for the triple (Max_Planck, placeOfBirth, Kiel), the vector representations of Max_Planck would be similar Albert Einstein's, Kiel's representation similar to Ulm. Thus placeOfBirth may also be detected as a synonym of the other relationships.

Our synonymous relationship detection technique makes use of this property by employing vector similarity as a measure for semantic similarity of relationships, whereas very similar vectors with a similarity larger than a certain threshold are likely to be semantically synonymous relationships. For measuring the semantic similarity between the relationship embeddings of vectors and matrices, we use standard vector norms. We have evaluated our method on the cosine similarity measure and on the L1-norm which is a distance measure. Note that a vector similarity of 1 means that two vectors are highly similar. Analogously, the vector distance of 0 implies high similarity. Cosine similarity is defined as $sim(r_i, r_j) = \frac{r_i \cdot r_j}{||r_i|| ||r_j||}$. It ranges from -1 to 1. The L1-norm is defined as $dist(r, r') = \sum_{i=1}^{d} |r_i - r'_i|$, d being the number of dimensions of the embedding. In contrast to cosine similarity, this norm is not restricted to a fixed interval, but is at least 0. If relationships are represented as a matrix, the entry-wise measures are computed. Computing the entry-wise measures of a matrix boils down to concatenate the columns of a matrix resulting in one large column vector. We use these similarity metrics for classifying relationship pairs as synonymous in the next step.

4.2 Classification of Synonymous Relationships

Finding synonymous relationships may be seen as a binary classification problem for some pair of relationships, where we have to separate synonyms from non-synonyms, based on their similarity. In the ideal case where knowledge embeddings can perfectly represent the semantics of a KG, very similar relationship representations imply that the relationships are synonym. For KG consolidation we need to classify all possible combinations of relationship pairs. Classification in our scenario is about determining a similarity or distance/similarity threshold for each relationships such that it separates synonymous from non-synonymous relationships.

As the first step, we compute a similarity histogram for every single relationship measuring its similarity/distance to all other relationships in the respective KG. Subsequently, we describe our method only based on distance metrics. However, the method is analogously used for similarity metrics.



(a) (b) (c)

Fig. 1. (a) A histogram with clear outlier for the relation `award ceremony`, (b) without any outlier `friend` and (c) with very similar relationships, but without an explicit outliers `title`.

In Fig. 1, we provide three exemplar histograms that we have built from a TransE model on the FB15K dataset from Freebase. The more left a relationship is located in the histogram, the smaller its distance to the respective relationship and the higher its semantic similarity. In Fig. 1(a), the majority of the relationships have an L1 distance of 6, whereas a single relationship has a distance of only 2. This relationship is seen as a clear outlier on the left side of the mass of the distribution. Hence, its vector distance is drastically smaller and its semantic similarity should be much higher. Indeed, this outlier is a synonym.

In contrast, we cannot find such an outlier in Fig. 1(b). Here, the histogram's mass has an average distance of 8. Outliers in this histogram may only be found on the right side of the distribution, being extremely dissimilar. The minimum distance of any relationship is at least 7. And indeed the respective relationship does not have any synonyms within our dataset.

For some relationships, outliers are not that easy to identify. In Fig. 1(c) for example, the most similar relationship has a distance of 2. Due to this variety in similarity histograms, a static and global threshold valid for all relationships of a KG is not suitable for this classification task. Instead, we aim at computing a dynamic threshold individually for each relationship based on outlier detection. Actually the relationship from Fig. 1(c) has several synonyms, but they can hardly be separated from the remaining relationships. It turns out that outliers usually are synonymous relationships, but not all synonymous relationships can be clearly identified as outliers.

In the second step, we perform the actual classification on these relationship-specific histograms. Since the similarity distribution usually are hardly skewed, we rely on an outlier detection based on the Z-score [23]. Given a similarity histogram for relationship r_i, we compute a Z-score for all (r_i, r_j), where r_j is another relationship from the KG. The Z-score is defined as: $z_{ij} = \frac{dist(r_i,r_j)-\mu_{r_i}}{\sigma_{r_i}}$, μ_{r_i} being the arithmetic mean and σ_{r_i} the standard deviation. Since the Z-score detects outliers based on their distance in terms of standard deviations from the arithmetic mean of the distribution, a fixed Z-score is used for classification of very diverse similarity histograms. With varying thresholds for the Z-score we can either achieve very precise results with low thresholds, or recall-oriented results with high thresholds.

In practice, similarity histograms for relationships have several outliers which sometimes can hardly be distinguished from the rest of the distribution, which makes a classification only based on the histogram very difficult. In these cases however, even a manual binary classification is extremely difficult and cannot be performed without detailed background knowledge. Further details are discussed in the evaluation section.

5 Evaluation

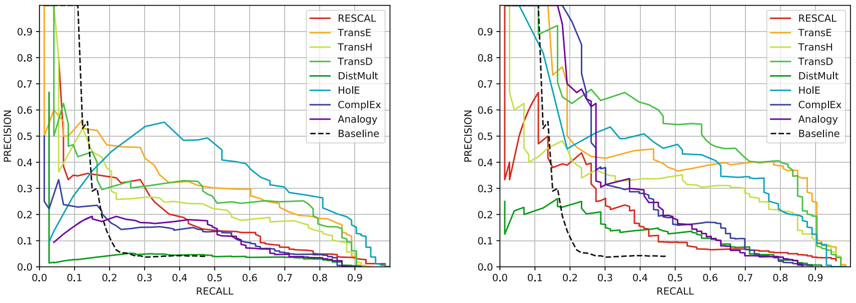
In the experiments, 8 different knowledge embeddings on several real-world KGs are trained and compared to the method from Abedjan et al. from [2], which is used as a baseline. We employ the knowledge embeddings RESCAL, TransE, TransH, TransD, ComplEx, DistMult, HolE and ANALOGY on Wikidata, Freebase and DBpedia. Additional results for other parameters, diagrams, datasets and scripts for reproducing the results may all be found in our Github repository[3]. Our implementation of the knowledge embeddings is based on the framework OpenKE [8] which comprises 9 knowledge embedding models. TransR [15] is excluded from the evaluation, since it was not able to return any synonymous relationships at all. The implementation of our classification, the evaluation scripts and the baseline systems are in Python.

In this section, we wanted to evaluate synonym detection in a two-fold manner: (1) Experiments where we could evaluate precision and recall with synthetic synonyms, (2) but also a real-world scenario where we are not making any assumptions when generating synthetic synonyms.

Overall this resulted in three experiments:

1. We first experimented on a subset of Freebase (FB15K [6]) that is known to perform very well for training knowledge embedding models. To measure recall and precision, synthetic synonymous relationships are introduced into Freebase.
2. The second experiment is performed on synthetic synonyms in Wikidata. A KG that has due to its size and sparseness rarely been tested for knowledge embeddings. Since Wikidata's size is not suited for knowledge embeddings to be trained on, a special sampling techniques that still allows to find all synonymous relationships is used.
3. The third experiment on DBpedia, a manual evaluation of the *Precision@k* for a large sample of DBpedia, instead of introducing synthetic synonymous is performed. In contrast to Wikidata, DBpedia is much more heterogeneous because it comprises a larger number of relationships. A measurement of the recall is not suitable here, because no gold standard of synonymous relationships is available. Building a gold standard would require manually checking millions of possible synonym pairs.

[3] https://github.com/JanKalo/RelAlign.



(a)

(b)

Fig. 2. Precision-recall-curves for synthetic synonyms on freebase. (a) Results with cosine similarity (b) Results with L1-Metric

In a final discussion, a comparison of the different experiments is made and cases where our technique could not identify synonymous relationships are further discussed. The discussion will also present the advantages and disadvantages of the different models and provide guidelines for choosing the right model for synonym detection.

Baseline Based on Frequent Itemsets. In all experiments, the 8 embedding models are compared to the baseline technique from [2]. Since no implementation is available for the baseline system for synonym detection from [2], we re-implemented the *Range Content Filtering* and *Reversed Correlation Coefficient* as described in the paper. Further details on our Python implementation are available in our Github repository. However, the technique has a *minimum support* as an input parameter for the range content filtering step, which highly influences precision and recall. We performed a grid search on the minimum support to tune this parameter to achieve highest F1 measure.

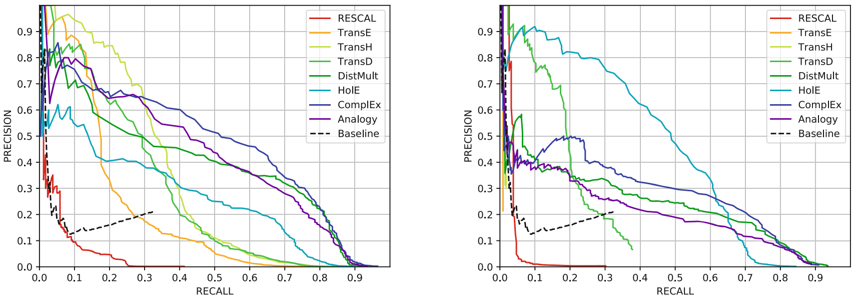
Synthetic Synonyms Generation. Synthetic synonyms are created by replacing relationship URIs with new (synthetic) URIs in existing triples of the dataset. As an example, we replace the triple (Albert Einstein, award, Nobel_Prize) with the triple (Albert Einstein, award_synonym, Nobel_Prize). `award` and `award_synonym` now have the identical meaning and are treated as synonymous relationships. To perform a proper relationship alignment task, the method has to re-identify these synthetic synonyms from the KG. For the synthetic synonym generation, an assumption from [2] is used so that the baseline can perform synonym detection. Abedjan et al. assume that synonymous relationships do not co-occur for the same subject entity. In case of our Einstein example, all triples about his awards would either use `award` or `award_synonym`, but should not mix the two for the same entity. This assumption stems from the idea that entities and their triples are often inserted at once by the same person or from the same data source. In such cases, synonymous relationships for the same entity are usually rare. For the experiments with synthetic synonyms, we introduced exactly one synthetic relationship for each relationship that occurs in at least 2000 triples and replaced it in 50% of the triples resulting in a 50-50 distribution of synonyms to non-synonyms. The F1-measure for all methods, including the baseline method, decreases the more skewed the distribution is, since it leads to some relationships being extremely rare, which negatively influences the embedding representation of a relationship. Results for the skewed distributions may also be found in our Github repository.

Sampling Method for Large Knowledge Graphs. Knowledge embedding training involves a lot of computational effort, which is why it should be performed on a fast GPU. Typical GPUs are very restricted in their memory size, making it impossible to train models for complete KGs. Training embeddings for example on the complete Wikidata dataset on a CPU is technically possible, but is around 10–100 times slower (i.e., several weeks) and thus prohibitive. To overcome this issue, we came up with a sampling technique that covers all relationships of a KG, but only a fraction of all triples. We randomly selected entities with all their triples in such a way that we have similarly many triples per relationship in our random sample. This sampling method guarantees for the knowledge embeddings still to work, while having enough information about each relationship so that its semantics is correctly be mapped to the latent vector space.

5.1 Evaluation of Synthetic Synonyms in Freebase

In this experiment, we compared knowledge embedding-based synonym detection with the baseline system on a subset from Freebase (FB15K) that is usually used to evaluate knowledge embeddings on link prediction [6]. FB15K comprises 592,213 triples about 15k entities, using 1,345 different relationship types. The dataset does not contain any literals, hence only triples where subject and object are resources. Originally, FB15K is a small part of Freebase that was chosen for link prediction, because it comprises a lot of triples per entity and lots of entities per relationship. It has been shown that this dataset is particularly well suited for training knowledge embeddings, also leading to good results in other tasks like link prediction. Since no gold standard for the existing synonymous relationships in FB15K is available, we have introduced synthetic synonyms. Overall 74 synonymous relationships have been added to FB15K.

The results of 8 knowledge embeddings and the baseline are presented in Fig. 2. The baseline achieves it highest precision of 1.0 at a recall of 0.11, but then drops to a precision of 0.05. For the minimum support of 0.02 leading to the best F1 measure, the recall never exceeds 0.5. This implies that 60% of the synonyms are never found. A lower minimum support also negatively influences the precision. Our knowledge embedding based approach on the other hand is evaluated with cosine and L1 metric. For the cosine similarity in Fig. 2(a), the baseline performs best for low recall values, but for a recall above 0.2 all models but DistMult perform better than the baseline approach. The results quality is even better for most models with L1 metric in (b). TransD is best in synonym



(a)

(b)

Fig. 3. Precision-recall-curves for synthetic synonyms on Wikidata. (a) Results with cosine similarity (b) Results with L1-metric

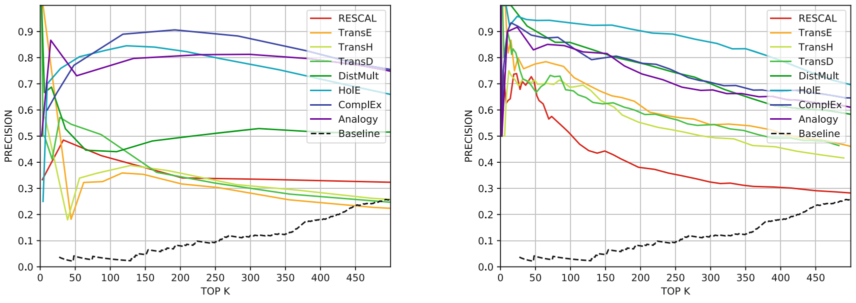
detection, achieving 1.0 precision at a recall of 0.1 and still 0.4 precision at a recall of 0.8.

Knowledge embeddings in this dataset achieve a high precision, for low recall values, but also find a lot of false positive synonymous relationships. These false positives are due to Freebase' fine granular modelling of relationships, leading to a high number of semantically very similar relationships that are not synonymous. Relationships in Freebase are defined for each entity type separately, implying that each relationship type is only used for a certain entity type. As an example several `genre` relationships are defined, depending on the class of the entities it is connected to. Differentiating `music_genre` from `film_genre` is quite difficult, but still possible with most embedding models. However, it gets even more difficult: FB15K contains 33 different `currency` relationships, all having a slightly different semantics, but very similar extensions. Hence being a problem for data-driven synonym detection techniques, when no background knowledge is given.

5.2 Synthetic Synonyms in Wikidata

The KG Wikidata is one of the fastest growing KGs that is openly available today. Our Wikidata version is from 9-19-2018. In contrast to other KGs, the Wikidata community is investing a lot of work into controlling its vocabulary. Therefore, it is supposed to be synonym free, which makes it a great candidate for evaluating our method with synthetic synonyms. Due to its size, we did not train knowledge embeddings on the complete Wikidata KG, but on a sample that comprises 15,663,641 million triples, with 341 synthetic synonymous relationships out of 1,797 relationships.

The precision and recall curves for all 8 models and the baseline are presented in Fig. 3. The knowledge embedding model-based approaches show a higher precision than the baseline for cosine similarity and L1-metric. Only RESCAL cannot hold up with any other system. The baselines starts with a high precision,



(a) (b)

Fig. 4. Manually evaluated Precision@k for synonyms in DBPedia. (a) Results with cosine similarity (b) Results with L1-metric

but sharply decreases and ends at a precision of 0.2 at a recall of 0.3. For the optimally chosen minimum support, the baseline only returns one third of all synonymous relationships. ComplEx and HolE achieve best classification results, outperforming the baseline by far. HolE has a precision of 0.75 at a recall of 0.3 and then is decreasing (cf. Fig. 3(a)). ComplEx in contrast is starting with a lower precision, but still has a precision of over 0.5 at a recall of 0.5 (cf. Fig. 3(b)).

Training good knowledge embeddings on a knowledge graph that is as sparse as Wikidata leads to lower quality models in contrast to FB15K, impairing the knowledge embedding quality. This also impairs the quality of synonym classification. However, Wikidata in contrast to FB15K does not contain highly similar relationships that could be misjudged as false positives by the classification technique. These two factors even out each other leading to a comparable quality to FB15K from the previous experiment.

5.3 Finding Synonyms in DBpedia with Manual Evaluation

As a last experiment, we also want to show that our method identifies existing synonyms in a large scale and very heterogeneous KG. Therefore, we evaluate our method with all embedding models and the baseline on a sample of DBpedia-16-2010. Due to its size, again a random sample similar to the procedure before is taken, resulting in a dataset with 12,664,192 triples and 15,654 distinct relationships.

For the manual evaluation on DBpedia, the annotator were supposed to evaluate relationship pairs into *synonyms* and *non-synonyms*. To measure the difficulty of the task, we first measured the inter-annotator agreement on a small sample of our dataset. We achieved an annotator agreement of over 0.90 for two independent raters, implying that the raters came to very similar results. Due to this experiment and due to the size of the dataset, we decided for only a single annotator for the manual evaluation. This manually build dataset stems from the top 500 results for each embedding model and the baseline summing up to around 3600 relationship pairs of which 1100 have been classified as correct. The dataset is also available in our GitHub repository. Now, we are able to obtain *Precision@k* values up to $k = 500$.

The results as *Precision@k* of our manual classification are presented in Fig. 4. For the baseline approach in this experiment, we chose a minimum support that returns around 500 results, so that it is comparable to the other results. Choosing a lower minimum support would increase the number of returned results, but decreases the precision. In contrast to the other models, the baseline starts with a low precision for $k = 50$, with a steadily increasing precision of up to 0.25 at $k = 500$. Note that the baseline is never exceeding a precision of 0.3 with the chosen minimum support value. The unconventional behaviour of the curve is due to Abedjan et al. making an assumption on the data that is not valid for DBpedia: They penalize synonymous relationships that co-occur for the same subject. The precision of our classification method on top of knowledge embeddings is showing higher precision for almost all models. HolE, ComplEx and ANALOGY all show comparably high precision values, also for high k values, whereas the translation embedding models TransE, TransD and TransH are quite weak in contrast to the earlier experiments. HolE with L1-metric in Fig. 4 show the best results with a precision of 0.94 at $k = 50$ and still a precision of 0.7 at $k = 500$.

During the extensive manual evaluation of the models, we got a detailed insight into the advantages and disadvantages of the models on DBpedia. Very frequent synonymous relationships that can clearly be distinguished from others manually are also clearly identified as synonyms by the embedding models. These are for example relationships for `genre`, `almaMater`, `deathPlace`, `birthPlace` and `award`. Problematic, at least in DBpedia, are rarely used relationships (`fuelSystem`, `drums`), relationships with spelling errors in their label (`amaMater`, `birthPace`) and relationships that are very similar to others other existing relationships (`club`, `youthteam`). Several other false positives stem from DBpedia containing relationships that are automatically extracted from external data sources that should be integrated and reformulated. As an example, DBpedia imports an external baseball database by creating two relationships for every row of a table with two columns: e.g. `stat1label`, `stat1value` for the first row and `stat2label`, `stat2value`. These false positives are not synonymous relationships, but obviously problematic relationships that should be reformulated.

5.4 Discussion of the Results

In all three experiments, we have shown the advantages of our embedding-based classification method on a variety of knowledge graphs. The baseline has been outperformed with almost all embedding techniques, because it heavily relies on synonym relationships to share object entities. In contrast, knowledge embedding based approaches are able to detect synonyms even though they do not share any subject nor object entities. As an additional drawback, the baseline needed parameter tuning for the minimum support value which was a difficult trade-off between precision and recall.

We have seen that a large part of synonymous relationships are detected in knowledge graphs, if they are frequently used. The semantics of very rare relationships can hardly be mapped to the knowledge embedding, hindering data-driven synonym detection mechanism. All embedding models show varying qualities across the different datasets, with HolE showing consistently good if not the best results, when choosing L1-metric. For most other models also L1-metric is also showing better results. Still no model was able to identify all synonymous relationships with high quality only based on the KG itself.

The fine-grained modelling of relationships (as in Freebase and DBpedia) is often problematic, since these relationships may hardly be distinguished from real synonyms, even in our extensive manual evaluation. We observed that relationship pairs that have been counted as false positives often are pairs of relationships that are extremely similar.

For example `/education/university/local_tuition./.../currency` and `/education/university/domestic_tuition./.../currency` both are highly similar in their extension, however are, semantically speaking, slightly different. One is used for the currency of the tuition at universities for local students and one for domestic students. We believe that these relationships could be integrated and the information about local and domestic students could be modelled differently. Such a difference cannot be observed by a purely data-driven approach.

6 Conclusion

In this paper, the suitability of the relationship representation in knowledge embeddings to measure semantic similarity between relationships is analyzed for the first time. We develop a new classification technique for identifying synonymous relationships for knowledge graph consolidation. In several large-scale experiments on Freebase, Wikidata and DBpedia we demonstrate how our classification method, employing a variety of existing knowledge embeddings, identifies synonyms with high precision and recall. Our approach does not make any assumptions on the data or labels of relationships. Thus, as our experiments have shown, our approach is generalizable to arbitrary knowledge graphs and is not depending on any additional domain-specific knowledge.

We showed that a traditional technique based on frequent item set mining [2] is not capable of competing with the presented classification method using relation embeddings from state-of-the-art relational learning techniques. The baseline approach was outperformed by almost all models on all datasets, since it returns several false positives. This has shown that identifying synonymous relationships indeed is a very difficult problem. Our manual evaluation has revealed that sometimes the semantics of relationships is even difficult to grasp for humans, so that the difference between synonymous relationships and highly similar relationships is hardly noticeable if detailed background or domain knowledge is missing. To overcome such difficulties, in previous experiments, we have also experimented with employing additional ontological information like `range` and domain predicates, to improve the results for synonym detection. However, it was hardly possible to use this information for finding synonyms, because KGs often lack domain and range information, and even in the few cases where this information was present, it was not enough to improve synonym detection.

Moreover, in our experimental results almost all positively classified synonymous relationships already have compatible ranges and domains, thus the added value would be negligible. We believe that even though current relational learning models are far from achieving perfect results for synonymous relationship detection, it will be difficult to perform much better using a purely data-driven approach without any external domain knowledge. Overall, our knowledge embedding-based knowledge graph consolidation techniques have shown good performance on a variety of different knowledge graphs. If the precision of our approach is not sufficient it still may be used in a semi-automatic fashion making the task much simpler.

For future work, we plan to combine our work with our previous work on transitivity of synonyms in instance matching problems [10,14]. Furthermore, our manual evaluation has shown that the results are very promising for correcting badly chosen relationships, or for identifying misused relationships in triples. We would like to investigate this application more thoroughly. It would also be interesting to further follow the idea of using relationship embeddings for query expansion.

References

1. RDF documentation from W3C. https://www.w3.org/RDF/
2. Abedjan, Z., Naumann, F.: Synonym analysis for predicate expansion. In: Cimiano, P., Corcho, O., Presutti, V., Hollink, L., Rudolph, S. (eds.) ESWC 2013. LNCS, vol. 7882, pp. 140–154. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38288-8_10
3. Algergawy, A., et al.: Results of the ontology alignment evaluation initiative 2018. In: CEUR-WS: Workshop Proceedings (2018)
4. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: DBpedia: a nucleus for a web of open data. In: Aberer, K., et al. (eds.) ASWC/ISWC -2007. LNCS, vol. 4825, pp. 722–735. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-76298-0_52
5. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, pp. 1247–1250 (2008)
6. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Advances in Neural Information Processing Systems, NIPS 2013, vol. 26, pp. 2787–2795 (2013)
7. Dong, X., et al.: Knowledge vault: a web-scale approach to probabilistic knowledge fusion. In: Proceedings of the 20th International Conference on Knowledge Discovery and Data Mining, SIGKDD 2014, pp. 601–610 (2014)
8. Han, X., et al.: Openke: an open toolkit for knowledge embedding. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2018, pp. 139–144 (2018)

9. Hertling, S., Paulheim, H.: DOME results for OAEI 2018. In: OM 2018: Proceedings of the 13th International Workshop on Ontology Matching Co-located with the 17th International Semantic Web Conference (ISWC 2018), Monterey, CA, USA, 8 October 2018, vol. 2288, pp. 144–151 (2018)
10. Homoceanu, S., Kalo, J.-C., Balke, W.-T.: Putting instance matching to the test: is instance matching ready for reliable data linking? In: Andreasen, T., Christiansen, H., Cubero, J.-C., Raś, Z.W. (eds.) ISMIS 2014. LNCS (LNAI), vol. 8502, pp. 274–284. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-08326-1_28
11. Jain, P., Hitzler, P., Sheth, A.P., Verma, K., Yeh, P.Z.: Ontology alignment for linked open data. In: Patel-Schneider, P.F., et al. (eds.) ISWC 2010. LNCS, vol. 6496, pp. 402–417. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-17746-0_26
12. Ji, G., He, S., Xu, L., Liu, K., Zhao, J.: Knowledge graph embedding via dynamic mapping matrix. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, ACL 2015, pp. 687–696 (2015)
13. Li, J., Tang, J., Li, Y., Luo, Q.: RiMOM: a dynamic multistrategy ontology alignment framework. IEEE Trans. Knowl. Data Eng. **21**(8), 1218–1232 (2009)
14. Kalo, J.C., Homoceanu, S., Rose, J., Balke, W.T.: Avoiding Chinese whispers: controlling end-to-end join quality in linked open data stores. In: Proceedings of the ACM Web Science Conference, WebSci 2015, pp. 5:1–5:10 (2015)
15. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, AAAI 2015, pp. 2181–2187 (2015)
16. Liu, H., Wu, Y., Yang, Y.: Analogical inference for multi-relational embeddings. In: Proceedings of the 34th International Conference on Machine Learning, ICML 2017, pp. 2168–2178 (2017)
17. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Proceedings of the 26th International Conference on Neural Information Processing Systems, NIPS 2013, vol. 2, pp. 3111–3119 (2013)
18. Nickel, M., Murphy, K., Tresp, V., Gabrilovich, E.: A review of relational machine learning for knowledge graphs. Proc. IEEE **104**(1), 11–33 (2016)
19. Nickel, M., Rosasco, L., Poggio, T.: Holographic embeddings of knowledge graphs. In: Proceedings of the 30th AAAI Conference on Artificial Intelligence, AAAI 2016, pp. 1955–1961. AAAI Press (2016)
20. Nickel, M., Tresp, V., Kriegel, H.P.: A three-way model for collective learning on multi-relational data. In: Proceedings of the 28th International Conference on Machine Learning, ICML 2011, pp. 809–816 (2011)
21. Nickel, M., Tresp, V., Kriegel, H.P.: Factorizing YAGO. In: Proceedings of the 21st International Conference on World Wide Web, WWW 2017, p. 271 (2012)
22. Pennington, J., Socher, R., Manning, C.D.: GloVe: global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, pp. 1532–1543 (2014)
23. Rousseeuw, P.J., Hubert, M.: Robust statistics for outlier detection. Wiley Interdisc. Rev. Data Min. Knowl. Discov. **1**(1), 73–79 (2011)
24. Suchanek, F.M., Abiteboul, S., Senellart, P.: Paris: probabilistic alignment of relations, instances, and schema. Proc. VLDB Endow. **5**(3), 157–168 (2011)
25. Suchanek, F.M., Kasneci, G., Weikum, G.: YAGO: a core of semantic knowledge. In: Proceedings of the 16th International Conference on World Wide Web, WWW 2007, p. 697 (2007)

26. Trouillon, T., Welbl, J., Riedel, S., Gaussier, E., Bouchard, G.: Complex embeddings for simple link prediction. In: Proceedings of the 33rd International Conference on Machine Learning, ICML 2016, vol. 48, pp. 2071–2080 (2016)
27. Vrandečić, D.: Wikidata: a new platform for collaborative data collection. In: Proceedings of the 21st International Conference on companion on World Wide Web, WWW 2012 Companion, p. 1063 (2012)
28. Wang, Q., Mao, Z., Wang, B., Guo, L.: Knowledge graph embedding: a survey of approaches and applications. IEEE Trans. Knowl. Data Eng. **29**(12), 2724–2743 (2017)
29. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. In: Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2014, pp. 1112–1119 (2014)
30. Weeds, J., Clarke, D., Reffin, J., Weir, D., Keller, B.: Learning to distinguish hypernyms and co-hyponyms. In: Proceedings of the 25th International Conference on Computational Linguistics: Technical Papers, COLING 2014, pp. 2249–2259 (2014)
31. Yang, Q., Wooldridge, M.J., Codocedo, V., Napoli, A.: Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, 25–31 July 2015 (2015)