Toward Measuring the Resemblance of Embedding Models for Evolving Ontologies

Romana Pernisch
University of Zurich
Switzerland
Vrije Universiteit Amsterdam
Discovery Lab, Elsevier
Netherlands
r.pernisch@vu.nl

Daniele Dell'Aglio
Aalborg University
Denmark
University of Zurich
Switzerland
dade@cs.aau.dk

Abraham Bernstein
University of Zurich
Switzerland
bernstein@ifi.uzh.ch

ABSTRACT

Updates on ontologies affect the operations built on top of them. But not all changes are equal: some updates drastically change the result of operations; others lead to minor variations, if any. Hence, estimating the impact of a change ex-ante is highly important, as it might make ontology engineers aware of the consequences of their action during editing. However, in order to estimate the impact of changes, we need to understand how to measure them.

To address this gap for embeddings, we propose a new measure called *Embedding Resemblance Indicator* (ERI), which takes into account both the stochasticity of learning embeddings as well as the shortcomings of established comparison methods. We base ERI on (i) a similarity score, (ii) a robustness factor $\hat{\mu}$ (based on the embedding method, similarity measure, and dataset), and (iii) the number of added or deleted entities to the embedding computed with the Jaccard index.

To evaluate ERI, we investigate its usage in the context of two biomedical ontologies and three embedding methods—GraRep, LINE, and DeepWalk—as well as the two standard benchmark datasets—FB15k-237 and Wordnet-18-RR—with TransE and RESCAL embeddings. To study different aspects of ERI, we introduce synthetic changes in the knowledge graphs, generating two test-cases with five versions each and compare their impact with the expected behaviour. Our studies suggests that ERI behaves as expected and captures the similarity of embeddings based on the severity of changes. ERI is crucial for enabling further studies into impact of changes on embeddings.

CCS CONCEPTS

• **Computing methodologies → Knowledge representation and reasoning**.

KEYWORDS

knowledge graph embeddings, embedding similarity, ontology evolution

ACM Reference Format:
Romana Pernisch, Daniele Dell'Aglio, and Abraham Bernstein. 2021. Toward Measuring the Resemblance of Embedding Models for Evolving Ontologies. In *Proceedings of the 11th Knowledge Capture Conference (K-CAP '21), December 2–3, 2021, Virtual Event, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3460210.3493540

1 INTRODUCTION

Ontologies like the Gene Ontology (GO) [25] change over time. They are updated by inserting new knowledge, removing outdated information, and updating wrong one. At the same time, the GO is used in subsequent tasks such as functional enrichment analysis [10]. Consequently, the result of the functional enrichment analysis over GO can yield a different result depending on the input version [10].

In this work, we consider the computation of embeddings[1] as the subsequent task over an evolving ontology or knowledge graph. Embeddings are rarely used alone, but rather power subsequent applications such as link prediction or recommender systems. In the biomedical domain, embeddings are used to discover new drug-disease associations or protein-protein interactions, among others [30]. A small change in the ontology, e.g., a new subclass link between already existing classes, can have a big impact on the inherent structure of the embedding. Whereas changes to, e.g. a description of an entity, will not have any impact on the embedding. With impact, we refer to the difference between the two embeddings calculated on two different versions of the same ontology. However, there is always a collection of changes between two versions of an ontology and not every change in the ontology leads to significant changes in its embedding model. Additionally, the learning of an embedding model consumes considerable amounts of resources. Consequentially, the indication of a big difference between the old embedding and the new one would signal the necessity for recomputation [18, 29]. The opposite would signal that little would be gained from learning a new embedding and that meanwhile the previously learned model can be used.

So far, the research on embeddings has focused on finding the best embedding algorithm, where "best" is measured in terms of link prediction or classification performance [30]. The comparison of the

K-CAP '21, December 2–3, 2021, Virtual Event, USA.

ACM ISBN 978-1-4503-8457-5/21/12…$15.00
https://doi.org/10.1145/3460210.3493540

[1] We will use the terms *embedding* and *embedding model* interchangeably throughout the paper to refer to the result of the learning process.

embedding model without a subsequent task is mainly known from natural language processing, where word embeddings are compared to determine drifts in languages using similarity measures [29]. Hence, little is known about how embeddings are influenced by the evolution of knowledge captured in the ontologies, which are used as input. *A natural and useful step forward is the investigation of impact measures able to warn ontology users of grave changes* [16, 29]. Thus, we first investigate the state of the art (SOA) on embeddings similarity measures and their suitability for comparing embeddings learned on different versions of the same ontology.

We identify their two shortcomings. First, they capture stochasticity of the embedding algorithms, which can be seen by running the same algorithm multiple times on the same data and observing that the resulting embeddings can largely differ. This leads to our main research question (RQ):

RQ: *How can shortcomings of embedding similarity measures be overcome?*

We propose a new impact measure called *Embedding Resemblance Indicator* (ERI), which is dependent on a similarity measure. ERI accounts for the stochasticity of the embedding method and corrects for the unmatched entities between the two embeddings. It captures the impact of changes, not the stability of embedding methods like similarity measures do. Hence, ERI signals how much an embedding changed because of the changes in the underlying KG used as input for the calculation.

To show that ERI behaves as intended, we develop test cases for which we generate synthetic knowledge graph versions. We apply the test cases to two well-established biomedical networks—drug-disease-associations (DDA) [21] and protein-protein-interactions (PPI) [23]—as well as on the FB15k-237 (FB) benchmark dataset [26] and Wordnet-18-RR (WN) [6, 13]. Additionally, we use five different embedding methods for them, choosing among the best and well known ones: GraRep [4], LINE [24], and DeepWalk [19] from the BioNEV package [30] for DDA, and PPI, TransE [28] and RESCAL [12] from the LibKGe Package [3] for FB and WN. All test cases confirm the usability of ERI, for the comparison of embedding models for an evolving input graph, ontology, or knowledge graph.

Therefore, our contributions are summarised as follows:

- The identification of shortcomings of embedding similarity measures and
- The introduction of ERI, an impact measure based on similarity measures, a robustness factor, and correction for applied changes.

Having an impact measure like ERI, which captures the changes in the embedding model without the stochasticity of the embedding method, is highly relevant for future research [29]. In the future, ERI could be learned, estimated, or approximated to inform ontology engineers and ontology users about potentially impactful changes between ontology versions. It can also enable research towards explainability of impact of changes on embedding calculations. Therefore, we present a very important step towards closing the communication gap between ontology engineers and ontology users by introducing an embedding impact measure.

Next, we introduce related work in different aspects of our research. This is followed by a section which identifies the shortcoming of SOA and defines ERI. We evaluate ERI in Section 4 in the context of a case study. Limitations and future work are addressed in Section 5 and Section 6 concludes our work.

2 RELATED WORK

There are different aspects related to the presented research. Ontology evolution is the stepping stone on which our research is based. *Ontology evolution* is a well studied and understood topic. Among others, Zablith et al. [31] survey various evolution processes and Hartung et al. [11] show the different tools for managing, exploring, and propagating changes on ontologies. Such studies focus on how ontologies are maintained. However, they exclusively look at the need for updates. Our interest lies in the consequences and effects of ontology evolution on downstream tasks.

A well-known consequence of evolution is *semantic drift* or semantic impact. SemaDrift [22] is a tool to calculate various semantic drift measures between versions of ontologies. It applies different methods of calculation and distinguishes between an exact, inexact, and hybrid ontology matching approach. In OntoDrift [5], the authors introduce an update to semantic drift measures used in SemaDrift for a better global comparison. In contrast to these tools, we compare the impact of ontology change in the context of embeddings, which can then be used in multiple more specific tasks like link prediction, classification, or recommender systems, instead of the ontologies directly.

There are other studies that directly investigate the effect of ontology evolution on a specific task. Gonçalves et al. [8] define a categorisation of changes based on entailment impact. For the classification of a change, they investigate if it influences the set of entailed axioms in the next version. Gross et al. [10] examine how the changes in an ontology impact previously conducted functional analysis. They define an impact measure based on stability of concepts over time and investigate the results of enrichment analysis using real and synthetic data to prove the effectiveness of the measure. Gottron and Gottron [9] also investigate the impact of KG evolution using LOD. They implement twelve different indexing methods and evaluate how the indices are affected by the evolution of the data using three different measures. Osborne and Motta [14] present the pragmatic ontology evolution, in which they analyze the selection of concepts for a new version by evaluating the performance of four different tasks. Our work, in contrast, focuses on the direct comparison of embedding models.

In order to study the impact of ontology evolution on embeddings, we need to compare the learned models. *Comparing graph embeddings*, the result of a stochastic calculation process, is a challenge. The most common way of evaluating an embedding model is through the performance in link prediction, graph completion, and entity classification tasks [30]. However, existing evaluations through a task are heavily dependent on the task itself, they are biased as well as unforgiving to false positives, which are not included in the test datasets [20, 30].

Comparing embeddings directly and absolutely is difficult: dimensions and distance between embedded entities have to be taken into account, which are dependent on a random seed. Looking beyond graph embeddings, there exist approaches to compare word embedding models in language processing [1, 27, 29]. Wang et al. [27] focus on biomedical datasets, and compare word embeddings

$$\begin{tikzcd}
E_h & E_i \arrow[rr, dash, dashed, bend left, "ERI(E_i, E_j, \hat{\mu}, sim(\cdot))", "sim(E_i, E_j)"'] & & E_j & E_k \\
O_h \arrow[u, "emb(O_h)"] \arrow[r] & O_i \arrow[u, "emb(O_i)"] \arrow[rr, "\delta^+_{i,j}, \delta^-_{i,j}"] & & O_j \arrow[u, "emb(O_j)"] \arrow[r] & O_k \arrow[u, "emb(O_k)"]
\end{tikzcd}$$

Figure 1: General model of the problem setting with ontology at time i (O_i), the changes ($\delta_{i,j}$) leading to O_j, embedding calculation ($emb(\cdot)$) result E_i and E_j.

based on medical texts. Wegmann et al. [29] focus on language models and investigate different types of semantic shifts in a specific language. They use local neighborhood (LN) similarity and global sentiment displacement to show that these two measures capture different types of semantic shift known in linguistics [29]. A visual approach has emerged implementing comparison methods and presenting neighbourhood distributions [2]. Since LN similarity is the most common comparison method, we focus on this approach and analyse its shortcomings.

Summarising, ontology/KG evolution and embedding comparison have been largely investigated. However, to our knowledge, this is the first work introducing a measure for the purpose of studying the *impact of ontology evolution on embeddings*.

3 THE EMBEDDING RESEMBLANCE INDICATOR (ERI)

This section first introduces the setting of impact between embeddings based on ontology versions. It then discusses established similarity measures and their shortcomings. Finally, it proposes solutions to these shortcomings and, based on those, we define ERI.

3.1 Setting

An ontology O is a set of triples (s, p, o), where s is the subject, p the predicate, and o the object. An entity has a uniform resource identifier (URI) and in most ontologies and KGs also a reference r for internal usage. Subjects are always entities and objects are either entities or literals. For some embedding methods, also literals receive a reference r but they do not have an identifier within the KG. However, most embedding methods take into account only entities[2].

We follow the terminology introduced by Pernischová et al. [18]. Let O_i and O_j be two different versions in O. The changes which lead from O_i to O_j are summarised with $\delta^+_{i,j}$ for additions and $\delta^-_{i,j}$ for deletions. We refer to set of additions and deletions with $\delta_{i,j}$. Analyses on the evolution of the National Cancer Institute Thesaurus and other biomedical ontologies, like those by Gonçalves et al. [7] and Pernisch et al. [15], suggest that this sets of changes can range from small (e.g. empty or one) to extremely large (e.g. 30'859 subclass axioms, 9'070 classes and 23 object properties as reported by Gonçalves et al. [7] for one specific version). Therefore, our definition does not restrict the number of changes $\delta_{i,j}$.

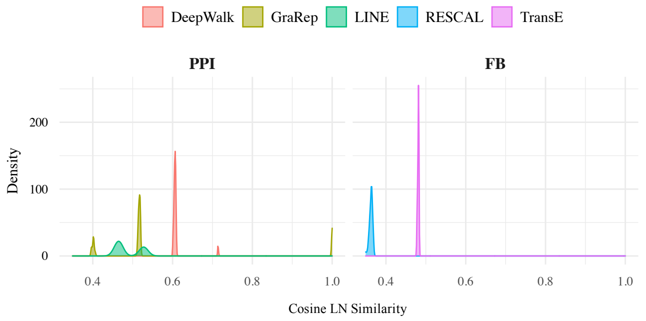


Figure 2: Distribution of intra-version similarity for PPI and FB (10 runs on O_{base}).

We define $emb(\cdot)$ as a function which applies to an ontology and produces the result E. In this framework, an embedding E is a set $\{e_1, \ldots, e_n\}$, where e_i denotes a pair (r_i, v_i), composed by the URI $r_i \in R$ identifying the entity, and the embedding $v_i \in V$ of r_i. A reference r uniquely identifies an entity from the ontology O.

The operation $emb(\cdot)$ takes as arguments an ontology and zero or more additional parameters if necessary. When the operation $emb(\cdot)$ is applied to O_i, it creates the result E_i. Given O_i and O_j, the respective results E_i and E_j can be the same (if the changes do not affect the result of $emb(\cdot)$), or they can differ.

3.2 Shortcomings of LN Similarity

We investigate the local neighborhood (LN) similarity as it is a widely accepted measure for comparing embedding models [18, 27, 29]. The LN similarity compares the neighbourhood of matching entities in two embeddings to each other. This means that only common entities between the embeddings ($R_i \cap R_j$) are compared. First, LN calculates the distance of the entity to all other entities in their embedding. Next, the Jaccard index is used to compare the set of closest neighbours between the two embeddings subject of the comparison. Depending on the usage, a number of closest neighbours is chosen, for example 25 or 100. Given these calculations the entities only present in O_i or only in O_j (i.e., $\forall r : (r \in R_i \lor r \in R_j) \land r \notin R_i \cap R_j$) are not taken into account. The comparison of embeddings only happens on common entities between the input ontologies, whose identifiers match ($R_i \cap R_j$). This is the first shortcoming of SOA embedding similarity metrics, especially metrics which take an identity-based approach. Therefore, we propose to add a set comparison of R_i and R_j to the embedding similarity. Capobianco et al. [5] take this approach to extend the calculation of semantic drift for the entire ontology. They use the Jaccard index to "correct" the calculated drift by the set comparison of entities between the compared ontology versions.

The second shortcoming concerns the comparison of results of a stochastic function. Two embeddings, calculated with different seeds, do not give the exact same result. Therefore, the similarity between models learned on the same version of the ontology (*intra-version*) captures the robustness of the embedding method. High similarity means that the algorithm delivers a reasonably similar model when executed multiple times. Low similarity shows that it is not providing stable models even when given the same input. A

[2] We ignore the embedding of properties in our definitions, as we do not require them in further definitions and calculations.

specific dataset and algorithm delivers a maximal similarity in an intra-version comparison that is rarely 1.

In Figure 2, we see the intra-version comparison for the Protein-Protein Interaction (PPI) network embeddings with DeepWalk, GraRep, and LINE, as well as the FB15k-237 (FB) with RESCAL and TransE. We can see that these values are distributed all around and only a few values of PPI's GraRep embeddings are 1. Especially for the embeddings of PPI, the quality of the embeddings and hence the results of the intra-version comparison differs greatly, as is visible by the multiple peaks for each algorithm. We do not see this for FB, however, those values are nonetheless far from 1. Given this comparison, the SOA similarity measures capture the robustness of an embedding method on a particular dataset, given the same input multiple times. With multiple calculations on the same input, we do not see a similarity close to 1. Therefore, similarity measures of this kind are not sufficient to capture the difference between embedding models, when the input graph changed (inter-version comparison). Hence, there is a need for a measure that regards the stochasticity of the embedding algorithm. We propose to use intra-version similarity as an upper bound of the achievable similarity given a dataset, embedding method and similarity measure.

3.3 Definition of ERI

As discussed above, similarity measures fail to consider two aspects: unmatched entities between two models and the stochasticity of the method. We propose to correct the similarity measure in two different ways to counteract these shortcomings and to receive a result that shows the difference of embedding models based on the changes between two ontology versions.

The first shortcoming of standard similarity measures is that they only compare entities which are present in both embeddings. We propose to follow the idea of [5] and use the Jaccard index. The Jaccard index is a set comparison measure, which we apply to the sets of references of each embedding model (R_i and R_j):

$$J(R_i, R_j) = \frac{|R_i \cap R_j|}{|R_i \cup R_j|} \tag{1}$$

When the Jaccard index is equal to 0, it indicates no overlap between the two sets of references and 1 means that all entities are the same. The comparison of embeddings needs to include the entire ontology, not just the semantic shifts, which is what similarity measures can capture according to Wegmann et al. [29].

We address the second shortcoming—the stochasticity—by proposing a correction of the similarity based on the robustness of the embedding method, an upper-bound given by the intra-version similarity of embedding models. We define $\hat{\mu}$ as the robustness of an embedding method. $\hat{\mu}$ is the mean of the similarity between embeddings learned on the same input ontology. The mean seems the best middle ground, according to the different distributions seen in Figure 2. Therefore, our definition of ERI includes $\hat{\mu}$, which characterizes the robustness of the embedding method and is dependent on the similarity measure, embedding algorithm, and the dataset input version O_i. With the addition of $\hat{\mu}$, our impact measure does not capture the stability of the embedding method but rather the changes in the underlying structure, making it orthogonal to similarity measures. For the calculation of $\hat{\mu}$, $emb(\cdot)$ is run N times on O, generating $E_i^{(k)}, k \in [1, N]$ on the base version E_i:

$$\hat{\mu}(sim(\cdot), E_i^N, N) = \frac{1}{N(N-1)/2} \sum_{n<m:n,m\in N} sim(E_i^{(n)}, E_i^{(m)}) \tag{2}$$

Putting these extensions together, we propose the following impact measure called *Embedding Resemblance Indicator*, short ERI, which overcomes the shortcomings of similarity measures:

$$ERI(E_i, E_j, \hat{\mu}, sim(\cdot)) = min(\frac{sim(E_i, E_j)}{\hat{\mu}}, 1) \times J(R_i, R_j) \tag{3}$$

where $J(R_i, R_j)$ is the Jaccard index defined in Equation (1), which accounts for the lack of comparison between new and removed entities using their references (r), $\hat{\mu}$ is the robustness factor defined in Equation (2). In this work, $sim(\cdot)$ is LN similarity, however, ERI could technically be used with any similarity measure.

The name for ERI refers to the capturing of how much two embedding models calculated on two versions of the same ontology resemble each other. The word resemblance suggests that ERI matches entities that are in common between two embeddings, but also accounts for the missing matches.

Discussion. Let us discuss the expected behaviour of ERI and consider a simple case of a taxonomy and its three versions: O_{base}, O_{add}, O_{del}. The difference between O_{base} and O_{add} is the addition of one leaf node and the difference between O_{base} and O_{del} is a deletion of one leaf node. Given our impact definition, the similarity between the embeddings learned on these three taxonomies is the same, contingent on a certain allowance for stochasticity of the embedding algorithm. The similarity between them will be 1, because the single difference between them is not taken into account by the similarity measure. We only compare the nodes that are common among O_{base}, O_{add}, and O_{del}. Using ERI, the Jaccard index (J) adjusts the similarity. Note that the adjustment would not be the same when comparing O_{base} and O_{add}, or O_{base} and O_{del}. The union of O_{base} and O_{add} has the newly added element, whereas the union of O_{base} and O_{del} would be missing that particular element, but would still contain all those from O_{base}. Therefore, the $impact(E_{base}, E_{add})$ and $impact(E_{base}, E_{del})$ values would not be equal because of J but close to each other, allowing for stochasticity of the embedding method.

Taking another case, let us consider O_{base} and O_{move}, where we move one node from a leaf to a node position. In this case, J is equal to 1, because all entities remain the same among O_{base} and O_{move}. The change is solely captured by $sim(E_{base}, E_{move})$ and should yield a bigger impact to the neighbourhood, because the entity has changed place in the hierarchy than the previous example.

Therefore, given a set of changes: a move of a node from leaf to root and the removal of a leaf, the move should have a larger influence on the impact measure than the removal of a leaf node. The removal of a root node would be trickier as the neighbourhood changes, but it is unclear how much from an intuitive point of view.

4 THE RESEMBLANCE CASE STUDY

In this section, we evaluate ERI based on a case study. We evaluate ERI on four datasets: Drug-Disease-Associations (DDA), protein-protein-interactions (PPI), Freebase FB15k-237 (FB), and WordNet-18 RR (WN). We selected appropriate embedding methods for DDA

Table 1: $\hat{\mu}$ and its standard deviation (SD) calculated with LN similarity, cosine distance, and 100 neighbours.

		number of embeddings compared			t-test p-value	
	Method	3	5	10	3vs5	5vs10
DDA	DeepWalk	0.5540 (0.2189)	0.5533 (0.2194)	0.5532 (0.2189)	0.597	0.732
	GraRep	0.4223 (0.2226)	0.4218 (0.2219)	0.4227 (0.2231)	0.779	0.417
	LINE	0.4397 (0.2330)	0.4411 (0.2332)	0.4409 (0.2331)	0.306	0.799
PPI	DeepWalk	0.6424 (0.2449)	0.6163 (0.2530)	0.6088 (0.2567)	0.315	0.348
	GraRep	1.0000 (0.0000)	0.8059 (0.0990)	0.6106 (0.1200)	0.002	0.003
	LINE	0.4864 (0.2401)	0.4896 (0.2393)	0.4857 (0.2387)	0.833	0.630
FB	RESCAL	0.3645 (0.1630)	0.3636 (0.1642)	0.3625 (0.1610)	0.128	0.063
	TransE	0.4806 (0.1923)	0.4811 (0.1924)	0.4810 (0.1916)	0.096	0.345
WN	RESCAL	0.1412 (0.1630)	0.1285 (0.1510)	0.1285 (0.1510)	0.001	0.732
	TransE	0.8244 (0.1429)	0.8231 (0.1433)	0.8125 (0.1466)	0.712	0.000

and PPI: GraRep [4], LINE [24], and DeepWalk [19], which are all part of BioNEV [30]. For FB and WN, we consider RESCAL [12] and TransE [28]. We picked TransE as it is a de-facto standard for embedding benchmarking, despite it is a translational method and it is not well suited for neighborhood similarity comparisons. We use the implementations of LibKGE [3].

4.1 Datasets and Tasks

For DDA, we extracted a drug-disease-association network from NDF-RT [21] version 2018/01/02. We followed the same extraction process as [30] using their code available online.[3] We extended the code with a parser from the RxNorm [32] format to OWL, which we provide on our website.[4] DDA is a bi-partite graph, where there are only connections between drugs and diseases. For PPI, we used STRING v11 [23] and their provided interaction for homo sapiens. We created a network using interactions higher than 800. The code is available as a Jupyter Notebook online (cf. Footnote 4). For the FB and WN datasets, we used the versions provided by LibKGE [3].

Before running the analysis, we ensured the embedding quality using a link prediction test, which is available in the framework provided by Yue et al. [30] and also Broscheit et al. [3]. Link prediction showed high performance of around 0.85 AUC, which we do not discuss further, since it is not subject of our case study or our research, but we provide the results in the project repository[4]. Finally, to evaluate ERI, we generated two sets of synthetic test cases (presented in in Section 4.3).

4.2 Estimation of $\hat{\mu}$

We take a closer look at $\hat{\mu}$ and the stability of the chosen embedding algorithms. To investigate the number of intra-version similarity comparisons necessary for $\hat{\mu}$, we generated ten embeddings for all datasets. In Table 1, we report $\hat{\mu}$ calculated using Equation (2) and the standard deviation for N equals 3, 5, and 10. This corresponds to $\frac{N(N-1)}{2}$ intra-version comparisons between the calculated embedding models. We use cosine LN similarity with 100 neighbors.

The results show that for most embedding algorithms and datasets, a small number of embedding calculations gives a reasonable estimation of $\hat{\mu}$. This finding is incredibly insightful for embedding methods, which require a long time for calculation as in the cases of TransE or LINE. When an embedding algorithm is performing well with link prediction, the variation in the intra-version comparison is not considerable and one does not need more than ten embedding calculations for a base estimation of $\hat{\mu}$. Dealing with a $\hat{\mu}$ as the one for WN with RESCAL (0.1285), the small difference is important because the comparison of inter-version embeddings is also below this value and a small difference in $\hat{\mu}$ matters for ERI. However, the value does not change between the intra-version comparison of five and ten embedding models.

We also see that especially WN shows smaller standard deviations. Similarly, FB with RESCAL and PPI with GraRep have a standard deviation below 0.2. All other embedding methods and comparison show values between 0.2 and 0.26. Table 1 also reports the p-value of a t-test to determine if the difference between the comparisons is statistically significant or not. We see that ten embeddings are not enough to achieve a stable $\hat{\mu}$ in the rare cases of GraRep with PPI and TransE with WN . Therefore, we use $\hat{\mu}$ calculated over ten embeddings in our evaluation of ERI. We calculated ten additional embedding and compared in total 20 embeddings. We found a $\hat{\mu} = 0.504$ ($sd = 0.230$) for GraRep with PPI and $\hat{\mu} = 0.810$ ($sd = 0.147$) for TransE with WN. The result with 3 embeddings for GraRep and PPI shows a similarity of 1. This does not mean that the embeddings are identical, since we are not comparing absolute numbers of the vectors but rather that the neighbourhoods are the same, when taking 100 neighbours into consideration. A t-test 10vs20 resulted in p-values of 0.000 and 0.099 respectively. For these two cases, we use $\hat{\mu}$ of 20 embeddings in subsequent calculations and 10 embeddings for all other dataset and embedding method combinations.

For all datasets, we used the base version for the calculations of $\hat{\mu}$. Since the estimation of the robustness factor already requires a considerable amount of embedding calculations, it would be beneficial if this estimation can be reused when the base version changes. We investigate this by calculating $\hat{\mu}$ for all test cases which we created (and explained in detail below). Table 2. As we can see, the choice of version the marginally influences the value of $\hat{\mu}$; however, we recommend to calculate $\hat{\mu}$ on the base version, so the older version of the two compared ontologies. We observe that the larger the difference between the versions, the larger the difference between the robustness factor $\hat{\mu}$. This shows that the embedding calculation and its stochasticity is not only dependent on the parameters of the algorithm but also on the input dataset.

$\hat{\mu}$ and the standard deviation show us how stable the calculation of an embedding given a particular method and dataset is. It is interesting to see that the big datasets (FB and WN) are slightly more stable than PPI and DDA. Unlike RESCAL and TransE, the three embedding methods executed on PPI and DDA do not calculate embeddings for properties. RESCAL and TransE also deliver an embedding of properties, which we do not regard in this evaluation because the LN similarity does not consider them. However, it is possible that they aid in providing a more stable embedding.

4.3 Test Cases

For our case study, we define two test cases based on the DDA dataset, because we are dealing with a bi-partite graph as input to the embedding method which is the most restricting of the four graphs. Here, bi-partite means that all relations in the dataset are

[3] https://github.com/xiangyue9607/BioNEV

[4] https://gitlab.ifi.uzh.ch/DDIS-Public/chimp-emb, which also includes all embeddings.

Table 2: $\hat{\mu}$ calculated using different versions (10 embeddings each). $\hat{\mu}$ calculated with base below method.

			v1	v2	v3	v4	v5
DDA	**DeepWalk**	TC 1	0.4961	0.5729	0.5657	0.5597	0.5968
	0.5532	TC 2	0.5845	0.5701	0.5585	0.5713	0.5613
	GraRep	TC 1	0.4306	0.4284	0.4289	0.4283	0.4429
	0.4227	TC 2	0.4281	0.4280	0.4252	0.4253	0.4318
	LINE	TC 1	0.4467	0.4494	0.4477	0.3512	0.3888
	0.4409	TC 2	0.4505	0.4507	0.4405	0.444	0.4482
PPI	**DeepWalk**	TC 1	0.6316	0.6740	0.6853	0.5125	0.4811
	0.6088	TC 2	0.6146	0.6122	0.6066	0.6082	0.6094
	GraRep	TC 1	0.5213	0.5411	0.5726	0.4662	0.5727
	0.6106	TC 2	0.5312	0.5286	0.5285	0.5232	0.5229
	LINE	TC 1	0.4593	0.4357	0.5416	0.4238	0.4451
	0.4857	TC 2	0.4719	0.4644	0.4701	0.4721	0.4260
FB	**RESCAL**	TC 1	0.3688	0.3771	0.3792	0.3818	0.3987
	0.3625	TC 2	0.3550	0.3585	0.3529	0.3578	0.3508
	TransE	TC 1	0.4922	0.4926	0.4894	0.4907	0.4938
	0.4810	TC 2	0.4890	0.4904	0.4902	0.4899	0.4810
WN	**RESCAL**	TC 1	0.1440	0.1507	0.1600	0.1711	0.1812
	0.1285	TC 2	0.1365	0.1319	0.1255	0.1267	0.1075
	TransE	TC 1	0.8074	0.7818	0.7788	0.7491	0.7323
	0.8125	TC 2	0.8214	0.8154	0.7954	0.7988	0.7904

between drugs and diseases. All additional information, such as hierarchy between drugs or between diseases, is not part of the graph and of the embedding. Even though this is not the case for PPI, FB, and WN, we treat them the same as DDA and apply the same generation procedure to create test cases for a better comparison.

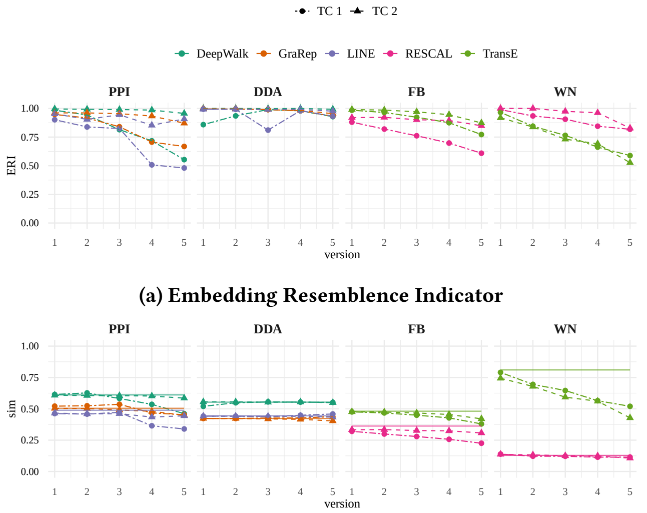
To show that ERI captures changes according to their severity, we generate two cases for testing. The first case (*TC 1*) is the deletion of edges between *low* degree entities. The second case (*TC 2*) is the deletion of edges between *high* degree entities. Edge deletions are solely based on the degree of nodes, and we focus on deletions because we do not want to inject non-existent and especially non-sensical data into the graphs. For each test case, we prepare five edge lists (v1 - v5), which differ in the number of to be deleted edges. Table 3 shows the number of deletions for each test case. DDA comes with 67'224 edges and PPI with 364'045 edges in their base networks used in these test cases. FB has 246'236 triples and WN has 86'726. A removal of an edge does not imply the removal of nodes, therefore, J = 1 when no nodes are removed in the edge deletion process. The corresponding code can be found in our repository[4].

For FB and WN, we ignore the types of properties and removed "edges" based on the degree of the entities they connect. We remove more edges for FB and WN than for DDA and PPI, because FB and WN are less well connected, and deleting edges between nodes with only one connection already lead to massive deletions, as it is reported in Table 3. We do not remove the property information, as it is required for the embedding methods, however we simply ignore it when removing connections between entities.

We observe the importance of the inclusion of the Jaccard index in the ERI. Given that TC 1 deletes edges between low-degree nodes, it is more likely that these nodes will end up without any connections to the rest of the graphs and will, hence, not be embedded. Therefore, we observe that the Jaccard index for TC 1 is lower than for TC 2. Traditional measures only focus on joint entities and do not measure the lack of overlap—a drawback ERI overcomes.

Table 3: Number of deleted edges and the Jaccard index for the synthetic versions of the two test cases.

			v1	v2	v3	v4	v5
DDA	TC 1	deleted edges	1	17	177	441	3'167
		$J(R_{base}, R_{v_i})$	0.96376	0.98757	0.93125	0.98030	0.93160
	TC 2	deleted edges	1	23	183	507	3'093
		$J(R_{base}, R_{v_i})$	1	0.99999	1	1	1
PPI	TC 1	deleted edges	127	1'111	15'415	29'667	74'292
		$J(R_{base}, R_{v_i})$	0.96342	0.89998	0.85518	0.70888	0.71569
	TC 2	deleted edges	106	1'148	15'846	30'542	74'294
		$J(R_{base}, R_{v_i})$	0.98777	0.97081	0.98777	0.96045	0.98600
FB	TC 1	deleted edges	1'322	5'167	13'280	25'648	54'528
		$J(R_{base}, R_{v_i})$	0.99601	0.99305	0.98879	0.98439	0.97930
	TC 2	deleted edges	1'623	5'156	13'943	27'467	57'348
		$J(R_{base}, R_{v_i})$	1	1	1	1	1
WN	TC 1	deleted edges	1'066	5'136	9'573	15'289	21'018
		$J(R_{base}, R_{v_i})$	0.98942	0.98456	0.95889	0.94993	0.91686
	TC 2	deleted edges	2'730	5'536	9'307	11'702	20'322
		$J(R_{base}, R_{v_i})$	1	0.99988	0.99978	0.99929	0.99665



(a) Embedding Resemblence Indicator

(b) Local neighbourhood similarity with cosine distance and $\hat{\mu}$. The legend from Figure 3a also applies here.

Figure 3: ERI and cosine similarity of case study. The straight full lines indicate $\hat{\mu}$ for each method and dataset.

As a consequence, we calculate ten embeddings for each generated edge list and dataset version. For the comparison between base and test case versions, we always report mean ERI to account for the stochasticity of the embedding process.

4.4 Results and Discussion

Figure 3 reports results of our case studies. In the figure, we can see ERI and the LN similarity calculations underneath each other. We report similarity and ERI with the cosine distance metric with 100 neighbours. We observe that as the version number progresses, the resemblance measured as ERI of embeddings is generally getting smaller. This behaviour is not as clearly visible with the similarity metric only. We also calculated ERI with euclidean and city-block distances, and they do not add additional findings. Therefore, we do not discuss them further for the sake of space and the calculated values are available in our repository.[4]

Figure 3b shows the similarity and $\hat{\mu}$ for the respective algorithms and test cases. $\hat{\mu}$ is the full line. As we can see, the similarity does not always capture any difference between the base similarity ($\hat{\mu}$) and testcases. This is especially true for WN with RESCAL, and with the DDA dataset, where $sim(\cdot)$ and $\hat{\mu}$ are practically constant across the versions. Also for FB, the similarity and the difference between the synthetic versions is minimal. Here, we can clearly see a benefit from ERI. Given that we have a difference in the number of entities, ERI reflects this using the Jaccard index in the comparison. However, that is not the only benefit. Using $\hat{\mu}$, we defined an upper bound for the similarity, and we can clearly see that this upper bound is not exceeded when comparing different input versions.

Even though we clearly observe the benefit of $\hat{\mu}$ and the Jaccard index, not all used datasets profit equally from ERI. On the one hand, WN with TransE already provided a promising comparison with the test cases without ERI, but with RESCAL that is not the case. On the other hand, DDA does not seem to benefit as much from ERI as other datasets. The values are close to 1, suggesting that the embedding is not experiencing much fluctuation even when many edges are removed. FB and PPI are clearly benefiting also between the two test cases, where TC 1 experiences more impact on the embedding than TC 2. This difference is less visible with the similarity metric and ERI makes it more apparent.

We believe that the answer of the height ERI for DDA lies in the structure of the dataset: DDA is a bi-partite network and the others are not. In DDA the "only" structural changes are edge removals between the two sets of nodes in the bi-partite graph. Hence, the bi-partite structure remains intact and learnable for the embedding algorithm, producing high ERI. In contrast, for PPI, FB, and WN, important structural elements might have been removed, leading to a larger impact on the embedding, and, consequently, experiencing lower ERI. It is also interesting to study the difference between TransE and RESCAL with FB and WN. Models for WN learned with RESCAL show less influence of the changes than those learned with TransE. For FB, we see the opposite: TransE is more robust against changes in input than RESCAL. We do not find differences between methods for PPI and DDA, which are as profound as with FB and WN. This shows that ERI reports on the changes to the underlying ontology, and not on the instability of the embeddings caused by the stochasticity of the embedding method. In an uncertainty scenario like the calculation of embeddings, it is unlikely to completely isolate such stochasticity. Therefore, we will never be able to attribute the difference in embedding models solely to the changes in the underlying ontology.

To summarise, the results show that our proposed impact measure overcomes the shortcomings of similarity measures. By taking established datasets and several well known embedding methods, we eliminate several uncertainties. Since we observe high performance with the link prediction for all datasets and algorithms, we are convinced that the learned embeddings in our evaluation are of high quality and can further be evaluated using ERI with confidence. Nonetheless, no matter how we choose to evaluate ERI, the embedding method's stochasticity can not be eliminated completely. Therefore, it is not possible to solely isolate the effect of the changes and how the impact measure captures them. However, we can approximate it with ERI. For ERI to be applied in a different scenario, we urge researchers to not forget the uncertainty of embedding calculations as such. ERI only considers the changes and we assume that researchers using embeddings in their applications are aware of the stochasticity of their method as such.

5 LIMITATIONS AND FUTURE WORK

Datasets and methods. We chose four well known datasets, which are often used for embedding evaluations. We also chose embedding methods, which are well established and known in the respective communities. The choice of methods was also based on the nature of the embedding methods and we covered the different approaches matrix factorisation, random walks, and neural networks. Therefore, we are confident that our evaluation generalises also to other datasets and embedding methods.

Similarity measures. The goal of this study is to define ERI, and not to investigate the meaning of the SOA similarity measures. We do not aim at categorizing these measures and advising on which aspects the similarity measures capture. Besides, ERI is not limited to the use of LN similarity measures: it provides the possibility to use other similarity approaches within its definition.

Robustness factor $\hat{\mu}$. Our robustness factor definition depends on the similarity measure and ontology, and not only on the embedding algorithm. We need further investigations on the stability of embeddings to determine to which degree $\hat{\mu}$ can be generalised and approximated. Currently, one first needs to calculate multiple embeddings to get an idea of the robustness factor, however that is not the case for every update of the ontology, as we have seen.

Evaluation. We use test cases, synthetic ontology versions of established datasets and embedding methods to assess whether ERI behaves as intended. We considered two specific domains where embeddings are used regularly. A measure like ERI is valuable and enables resemblance estimation in the future. One still needs to estimate the stochasticity of an algorithm over a specific dataset and similarity measure. Additionally, we used multiple embedding calculations to give a more rounded evaluation of ERI. However, in the real world scenario, applications do not operate on multiple embeddings, but on one. Hence, multiple calculations are warranted in this scenario, but would not be relevant when applied elsewhere.

As future work, we plan to evaluate embedding calculations and the impact of specific changes in more detail, taking into account the semantics of changes. ERI enables investigating different semantic changes and how they influence the learned model, which has not been possible so far due to the fact that SOA similarity measures are heavily limited by the stochasticity of the embedding algorithm. Our experiments suggest that even if the stochasticity can not be factored out completely, ERI is effective in considering it. However, stochasticity should not be dismissed and always be considered when using ERI to assess the evolution impact on an embedding.

6 CONCLUSIONS

The impact of ontology evolution is getting more attention as more ontologies are shared across the web and used in various applications. It is essential to consider the consequences of the growing and ever-changing knowledge that we, as researchers, capture in ontologies and use further [17, 29]. The consequences on the embedding have not been addressed before.

We found two essential shortcomings of SOA embedding similarity measures. We answer our research question by defining an impact measure called *Embedding Change Impact* (ERI). ERI builds on similarity measures and overcomes their shortcomings. ERI is orthogonal to similarity measures. It captures impact of changes but does not give indication on stability of embedding methods.

To evaluate ERI, we investigated its different aspects and show how it overcomes the shortcoming of similarity measures. We estimated $\hat{\mu}$, the robustness factor of the embedding method using different number of embeddings learned on the same version of the DDA network from NDF-RT [21] and PPI network from STRING v11 [23]. We found that with high quality embeddings, $\hat{\mu}$ can be estimated with a small number (e.g. five) of embedding comparisons. Using test cases, we generate synthetic ontology versions and calculate multiple embeddings. This process gives an estimation of the real impact of the changes on the embedding. The computed resemblance varies between the used datasets, but a comparison between the different test cases yields expected results. Therefore, we conclude that the proposed impact measure is valid for the application of embedding model comparison. We show that ERI overcomes the shortcomings of similarity measures—stochasticity of embeddings and disregard for changes to the ontology—but also has limitations.

This research is highly relevant, because it allows future investigation into the impact of ontology evolution on embeddings. With ERI, we can now investigate the learning and prediction of impact and informing ontology users about the necessity of recalculation as proposed by Wegmann et al. [29]. Approximating and/or predicting impact, like attempted by [18] but with similarity measures, would signal to ontology users if they need to update their current application and adapt to the new version. If the changes between versions were not enough to warrant the update, resources can be saved and used for the next update of the ontology and application. Depending on the application, only recalculating every other time could already be highly beneficial, if previously an update was made every hour. Hence, with ERI, we enable the research into this area, which could benefit many engineers, applications, and companies.

ACKNOWLEDGMENTS

We thank the Swiss National Science Foundation for their partial support under contract number 407550_167177. This work was also partially funded by Elsevier's Discovery Lab.

REFERENCES

[1] David Alvarez-Melis and Tommi S. Jaakkola. 2018. Gromov-Wasserstein Alignment of Word Embedding Spaces. In *EMNLP*. Association for Computational Linguistics, 1881–1890.

[2] Angie Boggust, Brandon Carter, and Arvind Satyanarayan. 2019. Embedding comparator: Visualizing differences in global structure and local neighborhoods via small multiples. *CoRR* abs/1912.04853 (2019).

[3] Samuel Broscheit, Daniel Ruffinelli, Adrian Kochsiek, Patrick Betz, and Rainer Gemulla. 2020. LibKGE - A knowledge graph embedding library for reproducible research. In *EMNLP: System Demonstrations.* 165–174.

[4] Shaosheng Cao, Wei Lu, and Qiongkai Xu. 2015. GraRep: Learning Graph Representations with Global Structural Information. In *ACM CIKM*. ACM, 891–900.

[5] Giuseppe Capobianco, Danilo Cavaliere, and Sabrina Senatore. 2020. OntoDrift: a semantic drift gauge for ontology evolution monitoring. In *MEPDaW@ISWC (CEUR, Vol. 2821)*. CEUR-WS.org, 1–10.

[6] Tim Dettmers, Minervini Pasquale, Stenetorp Pontus, and Sebastian Riedel. 2018. Convolutional 2D knowledge graph embeddings. In *32th AAAI*. 1811–1818.

[7] R. S. Gonçalves, B. Parsia, and U. Sattler. 2011. Analysing the evolution of the NCI Thesaurus. In *International Symposium CBMS.* 1–6.

[8] Rafael S. Gonçalves, Bijan Parsia, and Ulrike Sattler. 2011. Categorising logical differences between OWL ontologies. In *CIKM*. ACM, 1541–1546.

[9] Thomas Gottron and Christian Gottron. 2014. Perplexity of Index Models over Evolving Linked Data. In *ESWC*, Vol. 8465. Springer, 161–175.

[10] Anika Gross, Michael Hartung, Kay Prüfer, Janet Kelso, and Erhard Rahm. 2012. Impact of ontology evolution on functional analyses. *Bioinformatics* 28, 20 (2012), 2671–2677.

[11] Michael Hartung, James F. Terwilliger, and Erhard Rahm. 2011. Recent Advances in Schema and Ontology Evolution. In *Schema Matching and Mapping*. Springer, 149–190.

[12] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning Entity and Relation Embeddings for Knowledge Graph Completion. In *AAAI*. AAAI Press, 2181–2187.

[13] George A. Miller. 1995. WordNet: A lexical database for english. *Commun. ACM* 38, 11 (Nov. 1995), 39–41.

[14] Francesco Osborne and Enrico Motta. 2018. Pragmatic Ontology Evolution: Reconciling User Requirements and Application Performance. In *ISWC (LNCS, Vol. 11136)*. Springer, 495–512.

[15] Romana Pernisch, Daniele Dell'Aglio, and Abraham Bernstein. 2021. Beware of the hierarchy - An analysis of ontology evolution and the materialisation impact for biomedical ontologies. *Journal of Web Semantics* (2021).

[16] Romana Pernisch, Mirko Serbak, Daniele Dell' Aglio, and Abraham Bernstein. 2020. ChImp: Visualizing ontology changes and their impact in protégé. In *VOILA@ISWC*. CEUR-WS.org.

[17] Romana Pernischová. 2019. The Butterfly Effect in Knowledge Graphs: Predicting the Impact of Changes in the Evolving Web of Data. In *ISWC DC*. CEUR-WS.org.

[18] Romana Pernischová, Daniele Dell'Aglio, Matthew Horridge, Matthias Baumgartner, and Abraham Bernstein. 2019. Toward predicting impact of changes in evolving knowledge graphs. In *ISWC satellites*. CEUR-WS.org, 137–140.

[19] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. DeepWalk: online learning of social representations. In *SIGKDD*. ACM, 701–710.

[20] Catia Pesquita and Francisco M. Couto. 2012. Predicting the extension of biomedical ontologies. *PLOS Computational Biology* 8, 9 (Sept. 2012), 1–16.

[21] S. Rosenbloom, Joseph Awad, Ted Speroff, Peter Elkin, Russell Rothman, Anderson Spickard, Josh Peterson, Brent Bauer, Dietlind Wahner-Roedler, Mark Lee, William Gregg, Kevin Johnson, Jim Jirjis, Mark Erlbaum, John Carter, Michael Lincoln, and Steven Brown. 2003. Adequacy of representation of the National Drug. *AMIA* 2003 (2003), 569–78.

[22] T. G. Stavropoulos, S. Andreadis, E. Kontopoulos, and I. Kompatsiaris. 2018. SemaDrift: A hybrid method and visual tools to measure semantic drift in ontologies. *Journal of Web Semantics* (June 2018).

[23] Damian Szklarczyk, Annika Gable, David Lyon, Alexander Junge, Stefan Wyder, Jaime Huerta-Cepas, Milan Simonovic, Nadezhda Doncheva, John Morris, Peer Bork, Lars Jensen, and Christian von Mering. 2018. STRING v11: protein-protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets. *Nucleic acids research* 47 (2018).

[24] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. LINE: Large-scale information network embedding. In *WWW*. ACM, 1067–1077.

[25] The Gene Ontology Consortium. 2015. Gene Ontology Consortium: Going forward. *Nucleic Acids Research* 43, D1 (Jan. 2015), D1049–D1056.

[26] Kristina Toutanova and Danqi Chen. 2015. Observed versus latent features for knowledge base and text inference. In *CVSC*. 57–66.

[27] Yanshan Wang, Sijia Liu, Naveed Afzal, Majid Rastegar-Mojarad, Liwei Wang, Feichen Shen, Paul Kingsbury, and Hongfang Liu. 2018. A comparison of word embeddings for the biomedical natural language processing. *Journal of Biomedical Informatics* 87 (2018), 12 – 20.

[28] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge Graph Embedding by Translating on Hyperplanes. *AAAI* 28, 1 (June 2014).

[29] Anna Wegmann, Florian Lemmerich, and Markus Strohmaier. 2020. Detecting different forms of semantic shift in word embeddings via paradigmatic and syntagmatic association changes. In *ISWC*. Springer, 619–635.

[30] Xiang Yue, Zhen Wang, Jingong Huang, Srinivasan Parthasarathy, Soheil Moosavinasab, Yungui Huang, Simon M Lin, Wen Zhang, Ping Zhang, and Huan Sun. 2020. Graph embedding on biomedical networks: methods, applications and evaluations. *Bioinformatics* 36, 4 (2020), 1241–1251.

[31] Fouad Zablith, Grigoris Antoniou, Mathieu d'Aquin, Giorgos Flouris, Haridimos Kondylakis, Enrico Motta, Dimitris Plexousakis, and Marta Sabou. 2015. Ontology evolution: a process-centric survey. *Knowledge Engineering Review* 30, 1 (2015), 45–75.

[32] Kelly Zeng, John Kilbourne, Tammy Powell, and Robin Moore. 2011. Normalized names for clinical drugs: RxNorm at 6 years. *JAMIA* 18 (July 2011), 441–448.