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.aal.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 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

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 They are updated by inserting new knowledge, removing outdated information, and updating wrong one.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. Consequently, 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 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].

\(^1\)We will use the terms embedding and embedding model interchangeably throughout the paper to refer to the result of the learning process.
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 LibKG 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.
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 analyze its shortcomings.

Summarizing, 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 RESEMBLENCE 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 $\delta_{ij}^+$ for additions and $\delta_{ij}^-$ for deletions. We refer to set of additions and deletions with $\delta_{ij}$. 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 $\delta_{ij}$.

We define $\text{emb}(\cdot)$ 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

$^2$We ignore the embedding of properties in our definitions, as we do not require them in further definitions and calculations.
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 fail to consider two aspects: the comparison of embeddings needs to include the entire ontology, the Jaccard index is a set comparison measure, which we apply to the changes in the underlying structure, making it orthogonal to similarity measures. For the calculation of \( \hat{\mu} \), we propose the following impact measure called Embedding Resemblance Indicator, short ERI, which overcomes the shortcomings of similarity measures:

\[
ERI(E_i, E_j, \hat{\mu}, \sim(\cdot)) = \min\left(\frac{\sim(E_i, E_j)}{\hat{\mu}}, 1\right) \times J(R_i, R_j)
\]

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

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

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 \( R_j \)):

\[
J(R_i, R_j) = \frac{|R_i \cap R_j|}{|R_i \cup R_j|}
\]

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 \( \hat{\mu} \), \( \text{emb}(\cdot) \) is run \( N \) times on \( O_i \), generating \( E_i^{(k)}, k \in [1, N] \) on the base version \( E_i \):

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

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

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 fail to consider two aspects: the comparison of embeddings needs to include the entire ontology, the Jaccard index is a set comparison measure, which we apply to the changes in the underlying structure, making it orthogonal to similarity measures. For the calculation of \( \hat{\mu} \), \( \text{emb}(\cdot) \) is run \( N \) times on \( O_i \), generating \( E_i^{(k)}, k \in [1, N] \) on the base version \( E_i \):

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

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

\[
ERI(E_i, E_j, \hat{\mu}, \sim(\cdot)) = \min\left(\frac{\sim(E_i, E_j)}{\hat{\mu}}, 1\right) \times J(R_i, R_j)
\]

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 \( \sim(\cdot) \) and should yield a bigger impact to the neighborhood, 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 neighborhood 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 and PPI: GraRep [4], LINE [24], and DeepWalk [19], which are all
We created a network using interactions higher than 800. The code we provide on our website.


<table>
<thead>
<tr>
<th>Method</th>
<th>number of embeddings compared</th>
<th>t-test p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>DDA</td>
<td>DeepWalk 0.5540 (0.2189)</td>
<td>0.5533 (0.2194)</td>
</tr>
<tr>
<td></td>
<td>GraRep 0.4223 (0.2226)</td>
<td>0.4218 (0.2219)</td>
</tr>
<tr>
<td></td>
<td>LINE 0.4397 (0.2330)</td>
<td>0.4411 (0.2323)</td>
</tr>
<tr>
<td>PPI</td>
<td>DeepWalk 0.6424 (0.2449)</td>
<td>0.6163 (0.2530)</td>
</tr>
<tr>
<td></td>
<td>GraRep 1.0000 (0.0000)</td>
<td>0.8059 (0.0996)</td>
</tr>
<tr>
<td></td>
<td>LINE 0.4864 (0.2401)</td>
<td>0.4896 (0.2393)</td>
</tr>
<tr>
<td>FB</td>
<td>RESCAL 0.3645 (0.1630)</td>
<td>0.3636 (0.1642)</td>
</tr>
<tr>
<td></td>
<td>TransE 0.4806 (0.1923)</td>
<td>0.4811 (0.1924)</td>
</tr>
<tr>
<td>WN</td>
<td>RESCAL 0.1412 (0.1630)</td>
<td>0.1285 (0.1510)</td>
</tr>
<tr>
<td></td>
<td>TransE 0.8244 (0.1429)</td>
<td>0.8231 (0.1433)</td>
</tr>
</tbody>
</table>

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. Notebook 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 Broschelt 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 repository4. 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 $\mu$. 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 embedding 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 $\mu$ 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.

$\mu$ 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 relations between drugs and diseases. All additional information, such as
Table 2: $\hat{\mu}$ calculated using different versions (10 embeddings each). $\hat{\mu}$ calculated with base below method.

<table>
<thead>
<tr>
<th>Method</th>
<th>Version</th>
<th>TC 1</th>
<th>TC 2</th>
<th>TC 3</th>
<th>TC 4</th>
<th>TC 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepWalk</td>
<td>0.5532</td>
<td>0.9961</td>
<td>0.9972</td>
<td>0.9982</td>
<td>0.9992</td>
<td>0.9996</td>
</tr>
<tr>
<td>GraRep</td>
<td>0.4227</td>
<td>0.9036</td>
<td>0.8428</td>
<td>0.4283</td>
<td>0.4429</td>
<td>0.4276</td>
</tr>
<tr>
<td>LINE</td>
<td>0.4409</td>
<td>0.4567</td>
<td>0.4494</td>
<td>0.4477</td>
<td>0.3512</td>
<td>0.3888</td>
</tr>
<tr>
<td>DDA</td>
<td>0.6088</td>
<td>0.6316</td>
<td>0.6740</td>
<td>0.6853</td>
<td>0.5125</td>
<td>0.4811</td>
</tr>
<tr>
<td>GraRep</td>
<td>0.5106</td>
<td>0.5213</td>
<td>0.5411</td>
<td>0.5726</td>
<td>0.4662</td>
<td>0.5272</td>
</tr>
<tr>
<td>LINE</td>
<td>0.4857</td>
<td>0.4593</td>
<td>0.4357</td>
<td>0.5416</td>
<td>0.4238</td>
<td>0.4451</td>
</tr>
<tr>
<td>RESCAL</td>
<td>0.3625</td>
<td>0.3688</td>
<td>0.3771</td>
<td>0.3792</td>
<td>0.3818</td>
<td>0.3987</td>
</tr>
<tr>
<td>TransE</td>
<td>0.4810</td>
<td>0.4922</td>
<td>0.4926</td>
<td>0.4894</td>
<td>0.4907</td>
<td>0.4938</td>
</tr>
<tr>
<td>WN</td>
<td>0.1285</td>
<td>0.1140</td>
<td>0.1507</td>
<td>0.1609</td>
<td>0.1711</td>
<td>0.1812</td>
</tr>
<tr>
<td>TransE</td>
<td>0.8125</td>
<td>0.8074</td>
<td>0.7818</td>
<td>0.7788</td>
<td>0.7491</td>
<td>0.7323</td>
</tr>
</tbody>
</table>

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 lists (v1 - v5), which differ in the number of to be deleted edges. We do not remove the property information, because we do not want to inject non-existent and especially non-existent information.

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.

<table>
<thead>
<tr>
<th>Method</th>
<th>Version</th>
<th>TC 1</th>
<th>TC 2</th>
<th>TC 3</th>
<th>TC 4</th>
<th>TC 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDA</td>
<td>0.5968</td>
<td>1</td>
<td>17</td>
<td>177</td>
<td>441</td>
<td>3'167</td>
</tr>
<tr>
<td>GraRep</td>
<td>0.4227</td>
<td>0.9036</td>
<td>0.9757</td>
<td>0.93125</td>
<td>0.98030</td>
<td>0.93160</td>
</tr>
<tr>
<td>LINE</td>
<td>0.4409</td>
<td>0.4567</td>
<td>0.4494</td>
<td>0.4477</td>
<td>0.3512</td>
<td>0.3888</td>
</tr>
<tr>
<td>DDA</td>
<td>0.6088</td>
<td>0.6316</td>
<td>0.6740</td>
<td>0.6853</td>
<td>0.5125</td>
<td>0.4811</td>
</tr>
<tr>
<td>GraRep</td>
<td>0.5106</td>
<td>0.5213</td>
<td>0.5411</td>
<td>0.5726</td>
<td>0.4662</td>
<td>0.5272</td>
</tr>
<tr>
<td>LINE</td>
<td>0.4857</td>
<td>0.4593</td>
<td>0.4357</td>
<td>0.5416</td>
<td>0.4238</td>
<td>0.4451</td>
</tr>
<tr>
<td>RESCAL</td>
<td>0.3625</td>
<td>0.3688</td>
<td>0.3771</td>
<td>0.3792</td>
<td>0.3818</td>
<td>0.3987</td>
</tr>
<tr>
<td>TransE</td>
<td>0.4810</td>
<td>0.4922</td>
<td>0.4926</td>
<td>0.4894</td>
<td>0.4907</td>
<td>0.4938</td>
</tr>
<tr>
<td>WN</td>
<td>0.1285</td>
<td>0.1140</td>
<td>0.1507</td>
<td>0.1609</td>
<td>0.1711</td>
<td>0.1812</td>
</tr>
<tr>
<td>TransE</td>
<td>0.8125</td>
<td>0.8074</td>
<td>0.7818</td>
<td>0.7788</td>
<td>0.7491</td>
<td>0.7323</td>
</tr>
</tbody>
</table>

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 metrics.
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, \( \mu \), the robustness factor of the embedding method using different number of embeddings learned on the same version of the DDA network from NDF-RF [21] and PPI network from STRING v11 [23]. We found that with high quality embeddings, \( \mu \) 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—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 funded by Elsevier’s Discovery Lab.

REFERENCES


