University of Zurich[UZH]

Robustness of Drug-Disease-Association Network Embeddings

Thesis February 4, 2021

Tenzen Rabgang
of Zurich ZH, Switzerland

Student-ID: 11-490-398
tenzenyangzom.rabgang@uzh.ch

Advisor: **Romana Pernischova**

Prof. Abraham Bernstein, PhD
Institut für Informatik
Universität Zürich
http://www.ifi.uzh.ch/ddis

Acknowledgements

First and foremost, I would like to express my gratitude to my supervisor Romana Pernischova for her valuable input, time, and support throughout my thesis. I would also like to thank Prof. Abraham Bernstein for giving me the opportunity to write my master thesis at the DDIS group. Moreover, I would like to thank my sisters for advising and encouraging me, especially towards the end of the thesis. Last but not least, I would like to thank my parents for supporting me not only in my academic career but also throughout my whole life.

Zusammenfassung

Eine grosse Anzahl von Graph-Embedding-Methoden wurde bis dato vorgeschlagen, und mehrere biomedizinische Netzwerke haben vielversprechende Ergebnisse mit der Verwendung dieser Repräsentationen gezeigt. Mithilfe solcher Methoden kann jede Ontologie oder graph-ähnliche Struktur in eine niederdimensionale Vektordarstellung umgewandelt werden. Die Analyse von Graph-Embeddings über ein sich entwickelndes Netzwerk bleibt jedoch noch unerforscht. Daher verwenden wir 17 Drug-Disease-Association (DDA)-Graphen (Versionen) aus einem kontinuierlichen Netzwerk der gleichen Ontologie und setzen drei etablierte Embedding-Methoden ein. Unser Ansatz liegt darin, die Robustheit jeder Embedding-Methode über die Entwicklung der Ontologie hinweg zu bestimmen, indem wir die Ergebnisse aus zwei Anwendungsfällen analysieren und vergleichen. Zunächst führen wir einen Local Neighborhood Vergleich von Embeddings innerhalb derselben Version durch und vergleichen die Einheitlichkeit der Ergebnisse. In einem zweiten Schritt versuchen wir potenzielle Zusammenhänge zwischen Medikamenten und Krankheiten vorherzusagen. Hierzu vergleichen wir die Resultate der verschiedenen Versionen ebenfalls auf ihre Einheitlichkeit. Eine weitere Einschätzung zur Robustheit wird durch minimale Anpassungen eines Anwendungsfalles und folglich deren Einfluss auf das Ergebnis erzielt. Unsere Resultate zeigen, dass bestimmte Versionen in ihrer Entwicklung ein einheitliches Ergebnis liefern, und dass einige Embedding-Methoden stärker auf Veränderungen reagieren als andere.

Abstract

Graph embedding methods can transform any ontology or graph-like structure into a low-dimensional vector representation. An abundant amount of embedding methods have been proposed to date, and several biomedical networks have shown promising results with the use of such representations. However, the analysis of graph embeddings over an evolving network still remains unexplored. Therefore, we use 17 drug-disease association (DDA) graphs (versions) from an evolving network of the same ontology and apply three established embedding methods. Our approach is to determine the robustness of each embedding method across the evolution by analyzing and comparing the results of two application tasks. We first conduct a local neighborhood comparison of embeddings within the same version, then compare the results across the versions for consistency. Secondly, we use link prediction to find potential associations between drugs and diseases. Here, we compare the performance of each version to the others in order to prove consistency. In addition, we modify the parameters in a task to detect how sensitively the embeddings react to such a change and how it affects the task's result. This provides a further indication of the robustness of embeddings. Our findings demonstrate that certain versions in the evolution yield a consistent result, and some embedding methods react more strongly to parameter adjustments in a task than others.

Contents

1

Introduction

Ontologies and graph like structures are well established representations in biomedical research. One example can be found in the field of drug-disease associations (DDA) [1]. Associations between drugs and diseases are modeled as a graph and link prediction is a common task to analyze and find potential new connections which can then be explored further. To improve such predictions, a DDA network is embedded into a vector space. Previous research shows the feasibility and potential impactful findings using this method [32].

However, knowledge changes over time and so should the results of such down-stream tasks [14]. Therefore, we want to investigate, how much the performance of a certain task on a DDA data set changes over time. Or rather, how robust is the embedding and prediction against the evolution of the data set? To answer this question, we generate an evolving DDA data set from the National Drug File Reference Terminology (NDF-RT)[1] using the extraction method presented by Yue et al. [32]. We apply three embedding methods; two of them are included in the BioNEV[2] package [32] and the third one comprises an embedding method for bipartite graphs, introduced by Gao et al. [11]. We perform two tasks on the embeddings of each DDA network version and report the results. The first task consists of a neighborhood similarity comparison where we define multiple thresholds (distances) for the neighborhood. The second task is a link prediction method to forecast possible associations between drugs and diseases. Here, we strive to improve the prediction performance with logical inferences considering the domain context. Improving the model with logical inference brings us closer to the real world and reduces potential noise addition. We investigate the robustness of different embedding methods by analyzing and comparing the results of each task. The analysis of embeddings and link prediction over the evolution of the DDA network should answer the following research questions:

RQ1. How robust or consistent are the established embedding methods for DDA networks on a neighborhood similarity task with different distance (percentile) values across the evolution?

RQ2. How stable is the link prediction performance using established embedding methods for DDA networks against the evolution of the data set?

[1] *https://evs.nci.nih.gov/ftp1/NDF-RT/Archive/*
[2] *https://github.com/xiangyue9607/BioNEV*

To help answer the research questions above, we not only investigate the evolution of the DDA network, but also perform an experiment where we label two versions from the evolution as synthetic versions and define a third one as the ground-truth. We apply the above-mentioned tasks to find how the addition of noise affects the results and the robustness of the embeddings in this experiment. Hence, the introduced research questions are also applicable to this experiment.

The thesis is structured as follows: Chapter 2 reviews existing research about ontology evolution and embedded knowledge graphs. Chapter 3 covers a detailed explanation of the DDA data set with a short introduction of the next generation of NDF-RT. This is followed by Chapter 4, in which we introduce the embedding methods included in this work, along with an explanation of our approach and the data set we used. In Chapter 5, we present an initial visualization of the embeddings with a dimension reduction technique. In Chapters 6 and 7, we apply the embeddings on two tasks and present the results. Finally, the report is concluded with limitations, future works and a conclusion in Chapters 8, 9 and 10.

2

Related Work

There exists a plethora of research regarding graph embeddings. Since our work focuses on the performance of embeddings over an evolving ontology, we first study existing research in terms of ontology evolution. This is followed by an overview of works related to embeddings in the context of knowledge graphs (KG) and concluded by a review of available tasks to compare embeddings.

2.1 Ontology Evolution

Ontology evolution is a well examined research area and an abundant amount of works have been presented to date. Orme et al. [23] use several statistical metrics to analyze the data quality of ontologies, focusing on complexity and cohesion. Their metrics are inspired by object-oriented software metrics that measure design properties. They calculate the metrics for over 30 independent ontologies and align their conclusions to those of human evaluators. Further, they use several ontology instances from a single domain and examine the stability and completeness of evolving ontologies with respect to said metrics. Flouris et al. [10] conduct a literature review regarding changes in ontologies over several research disciplines and conclude that the boundaries between term usage/research area remain unclear. Consequently, they analyze and provide an explanation of terms as well as define the relationships between the research areas. They classify ontology changes into four groups: heterogeneity resolution, ontology modification, a combination of information from different ontologies, and ontology versioning. Gross et al. [14] analyze the impact of an evolving ontology with regard to subsequent statistical analysis, e.g. functional enrichment analyses. They define stability by comparing significant categories over two ontology instances. Two approaches are introduced; a basic approach, where categories are independently compared, and an advanced approach where categories are clustered semantically by distance and the number of overlapping category regions are matched. Pernischova et al. [25] take a step further where they estimate the impact of ontology changes. The purpose of this research is that KG engineers can estimate the impact of their actions beforehand, thus possibly preventing or breaking down anticipated changes or the addition of new knowledge. One case study involves the comparison of neighborhoods between two ontology instances, where they use the mean to measure the impact. Their results are promising with an Area under ROC curve (AUC) of 0.85 and therefore open another door for further research.

2.2 Graph Embeddings

Wang et al. [30] provide a survey of KG embeddings, where they distinguish between embeddings that rely on facts only and those that use additional attributes. By facts, the minimal information about a network is understood and, in this case, the relation between nodes. They recommend to work with models based on the open-world assumption (OWA) which states, that KGs contain facts and non-observed relations are either wrong or missing. Additionally they introduce translational distance models and semantic matching models. As the former already states, its scoring is based on distance, whereas the latter focuses on similarity regardless of distance. Kulmanov et al. [18] emphasize the limitations of graph embeddings and semantic similarity measures and therefore introduce EL embeddings, which are generated in Description Logics EL++[22]. The added value of using model-theoretic languages lies in the fact that semantic operators, e.g. conjunction or existential quantifiers, are also included. With the protein-protein interaction data set, they demonstrate that predictions are improved when using EL embeddings. Goyal et al. [13] state that most research related to graph embeddings focuses on preserving the node's characteristics in the graph, and little focus is given to evaluating the actual embeddings or comparison of different embedding methods. The authors claim that the following attributes determine the performance of graph embeddings: graph size, graph density, embedding dimension, and evaluation metric. They analyze several biological networks and discover an almost uniform distribution of densities (0.005-0.0155) in the graphs. Moreover, these graphs usually have small diameter ranges (8-12 or 16-18) and high clustering tendencies (clustering coefficient 0.10). Yue et al. [32] use several biomedical networks and apply 11 different embedding methods on them. Their research paper is especially important for our work as they also use the DDA data set and run different embedding methods on it. They perform a link prediction task with a Logistic Regression binary classifier, and achieve competitive performance. GraRep [3] yields the best performance with an AUC of 0.963, closely followed by struc2vec [27] and LINE [28]. Building on this, several computational methods have been introduced in recent years to identify associations between drugs and diseases. While traditional methods focus on including biological or chemical features in the prediction task [12, 19], graph embedding methods are promising to circumvent the possible lack of certain information. Dai et al. [7] propose a method based on matrix factorization to learn low-dimensional representations for drugs and diseases. Zhang et al. [35] use a similar approach, whereby they further introduce constraints such as similarities between drugs or diseases that can be added during factorization. In a previous study [34], the same authors proposed a neighborhood similarity method in order to find similarities between drugs respectively diseases. With this information, they used a label propagation process on a similarity-based graph to find associations between drugs and diseases. Generally, neighborhood similarity tasks have been widely used to compare embeddings [2, 15, 25]. Hamilton et al. [15] analyze different semantic shifts in two languages where one method introduced compares the nearest semantic neighbors. In [2], the authors compare the local neighborhoods of word embeddings and present an interactive tool that visualizes the neighborhood of such embeddings.

3

Exploratory Data Analysis

The NDF-RT published by the Veterans Health Administration (VHA) is an ontology used for modeling drug characteristics, including ingredients, chemical structure, dose form, physiologic effect, mechanism of action, pharmacokinetics, and related diseases. The core of the data set are the drug entities, which are among others associated with the related diseases. From 2009 until 2018, the VHA has periodically released new versions of the data set, leading to 82 releases in total. In the first section, the data set is introduced in detail, and the structure of the ontology is explained. In the second section, the data is analyzed, utilizing plots to visualize how it changed over the years since it first was published. Lastly, the continuation of NDF-RT is shortly introduced.

3.1 Data Description

The NDF-RT data set has its own ontological representation in XML format, derived from RxNorm [33]. Terms are defined as concepts and are hierarchically structured. Every concept is of a specific type and owns properties, roles, and associations. Furthermore, a unique alphanumeric identifier (NUI) is assigned to each concept, which is maintained across versions and can thus be used as a means for tracking and comparing. There exist eight different types of concepts. However, our primary focus lies on the following two: DRUG_KIND and DISEASE_KIND. Figure 3.1 illustrates the possible relations these two types can be associated with. The relations within and between the entities are very similar in their meaning; hence, we can presume a connection between them such as e.g., *may treat* pairs overlap with *may prevent*, or *may diagnose* pairs. However, as we have no medical background nor any biological knowledge, we will solely focus on the *may treat* relations and omit the rest.

The NDF-RT versions are publicly available in XML, whereby 20 versions also provide an OWL representation. For the extraction process, we use the logically inferred XML versions published in the same archive as the original. The inferred versions include the derived roles and associations of the concepts along the hierarchy. As a first step, we create a parser for the XML files to convert all 82 versions into an OWL format for better comprehension. The translation of the entities is inspired by the available OWL files, and Figure 3.2 shows the mapping between the respective attributes. An example

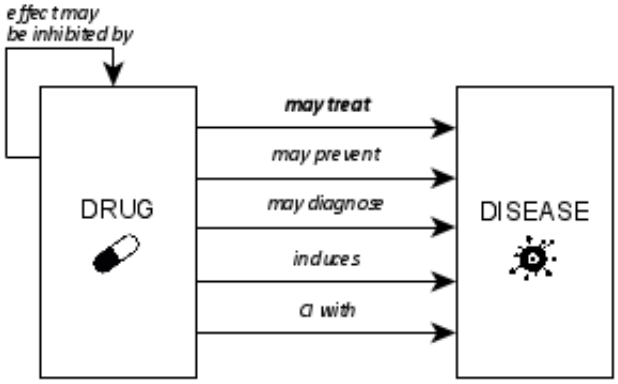


Figure 3.1: Drug-Disease relations (CI = contraindications)

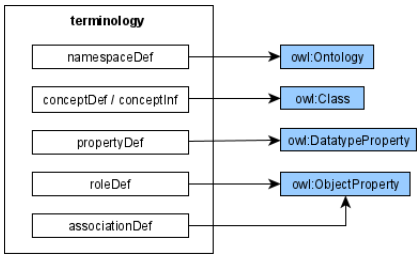


Figure 3.2: Parser for XML to OWL

of a drug entity and its relation to a disease in OWL format can be found in Appendix A.1.

After creating an OWL representation for each version, the next step is to extract the *may treat* relationships between the drug and disease entities and save them into an edge list. Each entity in the edge list is assigned to an integer (ID), which is further stored in a crosswalk file with reference to the NUI, the type (drug or disease), and the category (if available). Tables 3.1 and 3.2 provide an example of such mapping.

ID	NUI	Type	Category
1	N0000020115	Drug	[AM000] ANTIMICROBIALS
2	N0000020123	Drug	[OP000] OPHTHALMIC AGENTS
3	N0000000007	Disease	Eye Diseases [Disease/Finding]
4	N0000000265	Disease	Infectious Diseases [Disease/Finding]

Table 3.1: Node list structure

Drug ID	Disease ID
1	3
1	4
2	3
2	4

Table 3.2: Edge list with IDs from the node list

3.2 Data Statistics

For our research, we focus on drug and disease entities, and their *may treat* relationship. The resulting DDA graph is of heterogeneous nature or more specifically bipartite. The definition of a bipartite graph is that nodes can be grouped in two categories ($C1$ and $C2$) such that no edge connects nodes from the same category. In formal language, a bipartite graph is defined as $G = C1 \cup C2$, where $C1 = \{d_i \mid 1 \leq i \leq k\}$ and $C2 = \{s_i \mid 1 \leq i \leq j\}$ with $k =\mid C1 \mid$ and $j =\mid C2 \mid$ [11] as shown in Figure 3.3.

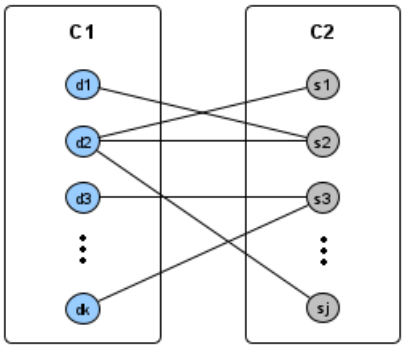


Figure 3.3: Example of a bipartite graph

Figures 3.4 and 3.5 demonstrate how the number of drugs and diseases changes over the years. The drug entities have been increasing steadily with peaks at the beginning of 2009 and the end of 2017. In contrast, the disease entities have remained static until the middle of 2017 at which point around 200 new diseases were added. This is expected as we assume the drug development rate to be much higher than the probability of discovering a new disease. Figure 3.6 depicts the number of links between drugs and diseases, and we observe a continuous growth similar to the drug history. This correlation is explainable since adding new drugs increases the number of links. Further, the peak at the end of 2017 is more prominent than the other peaks since not only drugs, but also disease entities were added to the ontology at that point. From this initial analysis, we can infer that the ontology's core content has grown continuously without any major changes over approximately ten years.

Next, we will take a closer look at the graph features of the drug-disease associations.

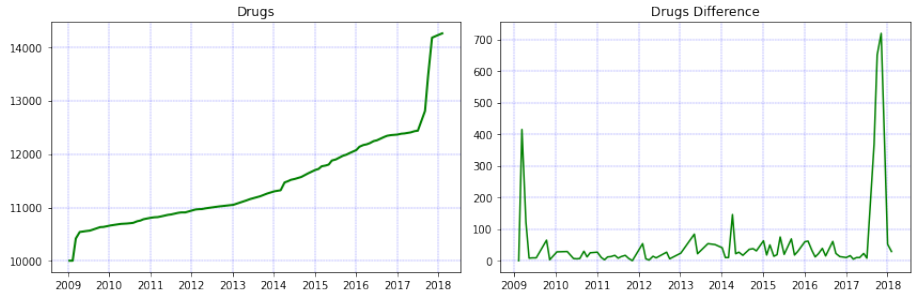


Figure 3.4: # Drugs across versions

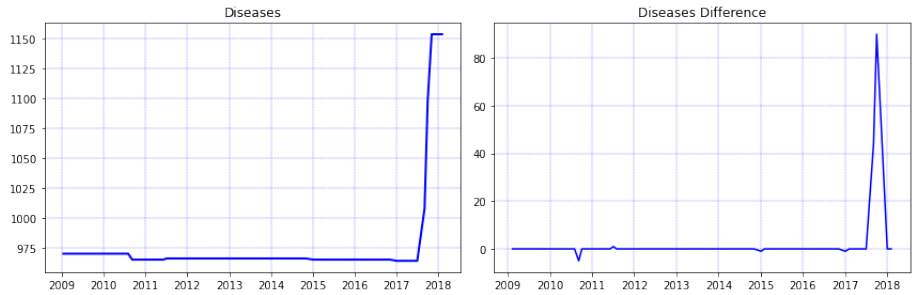


Figure 3.5: # Diseases across versions

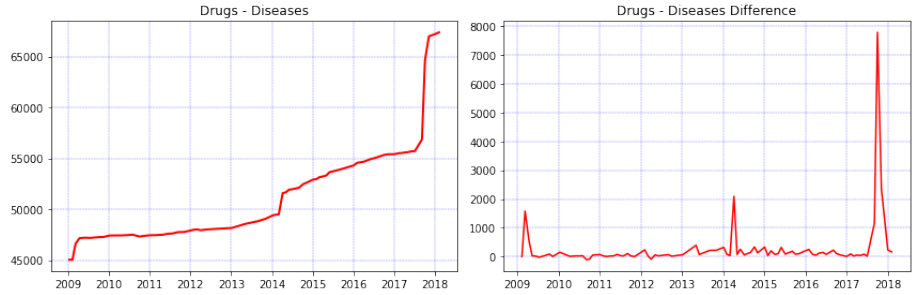
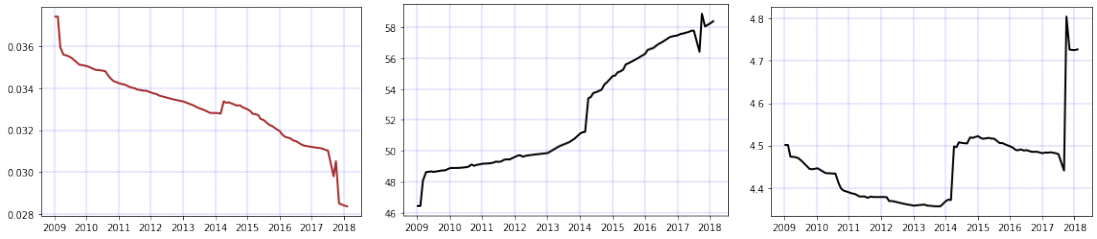


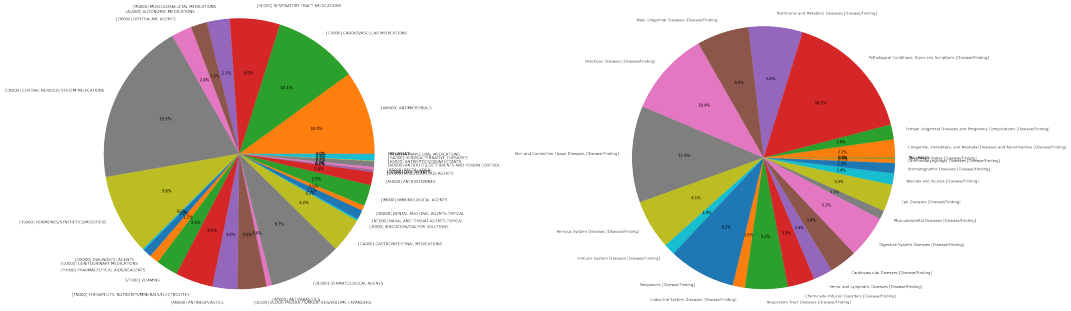
Figure 3.6: # Drug-Disease relations across versions

As already mentioned, biological graphs are known to be sparse, and the NDF-RT data set is no exception. In Figure 3.7, the left-most plot shows the density measure across versions. We notice a consistent decrease in density in the course of the evolution. This is expected, as we have seen previously, that drug or disease nodes are continuously added to the ontology, and with every new node, the number of possible links increases. The values overall are relatively low, ranging between approximately 0.02 to 0.03. However, this is not surprising as Goyal et al. [13] reported a density value of 0.005 to 0.0155 for several biological networks. The right two plots display the average node degree of the graph. We distinguish between node in-degree and out-degree values, and it is apparent that the number of in-degrees is much higher than the out-degrees. In other words, when



(a) Density in % (b) Average node in-degree (c) Average node out-degree

Figure 3.7: Graph statistics for NDF-RT



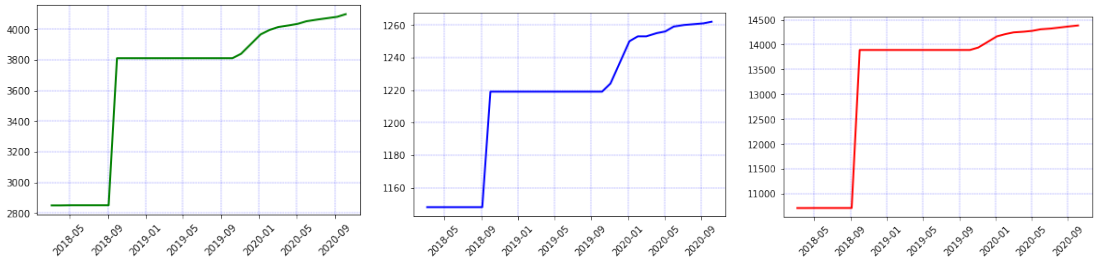
(a) Drugs (b) Diseases

Figure 3.8: Drugs and Diseases by categories

we start from the premise of a directed drug-disease graph, drug nodes have on average approximately four outgoing links while disease nodes hold around 50 incoming links. This difference in number can be explained by the higher amount of drug nodes than disease nodes. The steady increase in the in-degree values confirms the addition of links, and the stagnant out-degrees confirm the increase in the number of drugs.

Figure 3.8 shows the percentage of categories of drugs respectively diseases. In 3.8a, the largest category is visible with 20 percent (Central Nervous System Medications). The remaining lie around 10 percent or below and are evenly distributed over the available categories. The same applies to 3.8b; the largest category owns 16 percent, whereas the rest stays around 10 percent or lower. In the drugs chart, 20 percent of the entities are not included as these were incorporated from another source terminology, which did not label the drugs by category. On the other hand, the disease chart is complete because only one source, namely the Medical Subject Headings (MeSH) [26], is used. In total, there are 23 disease categories and 31 drug categories.

This initial analysis of the DDA data set gives an insight into the data structure and the severity of the changes across the evolution. Simultaneously, it acts as a guidance and auxiliary means when discovering unexpected behavior in the following chapters that are otherwise not explicable.



(a) # Drugs (b) # Diseases (c) # Drug-Disease relations

Figure 3.9: Changes in MED-RT across versions

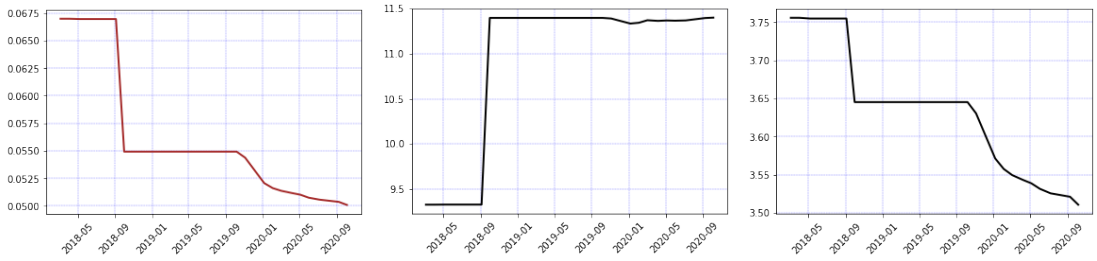
3.3 MED-RT

In 2018, the NDF-RT data set was replaced by the Medication Reference Terminology (MED-RT), which contains a much leaner data structure. Previously, concepts taken from external terminologies were incorporated into the NDF-RT terminology. However, in MED-RT they are newly referenced by a native unique identifier, their name, and the respective namespace (e.g. MeSH or RxNorm). Concepts defined and owned by VHA are further described in the MED-RT and labeled with NUI. Until now (October 2020), there exist a total of 27 releases.

We extract the drug-disease associations by generating an edge list and a node list file similar to the NDF-RT extraction. Due to time constraints, we are not able to build an OWL representation of the ontology and leave it as future works.

Next, we analyze the drug-disease associations similar to Section 3.2. Figure 3.9 presents the number of drugs, diseases, and drug-disease relations across different versions, and we notice a similar trend as in NDF-RT. The number of drugs is increasing faster than the diseases, and the drug-disease relations continue to grow. In all three plots, a peak at the end of 2018 and 2019 is clearly visible.

We further calculate the density of the graph, as well as the node in-degree and out-degree values, as presented in Figure 3.10. Here again, the density starts to drop as the ontology evolves, and the range lies a bit higher than in NDF-RT. The in-degree values initially increase up to 2 links but then remain static. The out-degree values remain static as the deviation stays in a one-decimal range. For all three plots, the declines, respectively peaks, occur at the end of 2018 and 2019, which coincides with the increase in drugs, diseases, and drug-disease relations.



(a) Density in % (b) Average node in-degree (c) Average node out-degree

Figure 3.10: Graph statistics for MED-RT

4

Embeddings

First of all, we introduce the basic notation for embeddings. A graph can be described as $G = (V, E)$, where V are nodes and E are edges. $N =\mid V \mid$ is equal to the number of nodes. In addition, edges can be represented as an adjacency matrix of size NxN, where if there exists an edge between v_i and v_j, then $e_{ij} = 1$ with $e_{ij} \in E$, otherwise $e_{ij} = 0$ with $e_{ij} \notin E$. The DDA data set represents the graph in this work and serves as the basis to create the embeddings. In this chapter, we present our approach on how we carried out the embedding generation and the DDA versions we used. In addition, we provide a short introduction of the applied embedding methods.

4.1 Approach

Since our focus lies in the evolution of an ontology, we select 17 DDA versions (every 5th from the 82 versions) and generate embeddings with dimension size 100 for 50 runs. The reason for selecting only a limited number of versions is primarily because of the embedding generation process. Creating embeddings is known to be computationally expensive, and due to the restricted time frame, we decide not to include all versions. Moreover, only minor updates were made in some consecutive versions, thus resulting in a very similar embedding and, therefore, evaluation. We describe this as the evolution-oriented approach, which we will refer to in the following chapters.

As an additional experiment, we determine the robustness of the embedding methods by adding noise to an original graph. Therefore, we choose one version (2014.06.02) as the ground truth and select two versions (2009.01.03 and 2018.01.02) as synthetic ones with different noise levels. To compare them, we first remove all nodes that do not appear in the ground-truth version. The same applies for the synthetic versions, where we remove nodes that exist in the ground-truth but not in the synthetic versions. In the end, the ground-truth and the synthetic versions comprise the same set of nodes but with different edges. We generate ten embeddings of the ground-truth and the synthetic versions. The noise addition for this experiment consists of validated edges that were added/removed in the previous and later version. We define two noise levels where 3% noise is added at level 1 and 10% at level 2. This experiment allows us to analyze how the addition or removal of edges affect the embeddings' performance in the two tasks. Table

	Ground-truth	Noise L1 (2009.01.03)	Noise L2 (2018.01.02)
Nodes	10'858	10'858	10'858
Edges	45'825	44'543	50'803
Noise (in edges)	-	1'282 (+1'066 / -2'348)	4'978 (+5'861 / -883)

Table 4.1: Noise addition to version 2014.06.02

4.1 presents a short summary. Similar to above, we will refer to this as the noise-oriented approach in the following chapters.

To summarize, we presented the evolution-oriented approach that consists of 17 DDA versions with 50 runs and the noise-oriented approach with 10 runs. For the tasks in Chapter 5 and 6, we proceed as follows: first, we take an evolution-oriented perspective, where we analyze and report the results with an evolving ontology. Second, we convert to a noise-oriented perspective, where we take one version and consider the previous and future version as noise in the ontology.

4.2 Methods

There exist countless methods for embedding generation, which can be roughly divided into the following three categories: matrix factorization (MF), random walks, and neural networks (NN). As previously mentioned, Yue et al. [32] conducted a comprehensive analysis of several embedding methods. One of the used data sets is a DDA network released in March 2017. They apply 11 embedding methods followed by a link prediction. The best-performing methods for the prediction task are GraRep [3], LINE [28], and struc2vec [27]. In this work, we choose GraRep and LINE from the BioNEV[1] package in order to compare an MF-based method with an NN-based method. As struc2vec is computationally too expensive, we select an alternative embedding method named BiNE [11], which is based on random walks and specifically constructed for bipartite graphs. It is important to mention that the applied methods can only generate embeddings from known nodes. To elaborate, these methods only allow an edge list containing nodes with at least one edge as an input while nodes without edges are omitted. Therefore, we have to keep in mind that the analyses and evaluations are performed in a controlled environment, which does not allow any interference.

GraRep is an MF-based method that preserves global structural information of a graph with the k-step algorithm. Every k-step model outputs a local representation of a node, and by concatenating those, we obtain a global representation of a node. Hence, this method focuses on the neighborhood structure of each node, such that nodes close to each other have similar embeddings and vice versa. LINE is an NN-based method that integrates both local and global network structures in an embedding. It focuses on the first- and second-order proximity, and is therefore, similar to GraRep, prioritizing the neighborhood when learning the embeddings. Both methods believe

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

that including neighborhood-related information in the embedding process will preserve the global structural information of a graph. The hyperparameters are taken from Yue et al. [32] as they already conducted an in-depth tuning.

BiNE is an embedding method explicitly built for bipartite graphs and is not provided in the BioNEV package. This method distinguishes between explicit and implicit relations of a graph, in other words, between first- and second-order proximity. For the latter, the bipartite graph is divided into two homogeneous graphs (drug-drug and disease-disease graphs) and random walks with a biased, self-adaptive algorithm are used. In Gao et al. [11], hyperparameter tuning is conducted by measuring the impact on link prediction. They tune the parameters β and γ as these are crucial indicators for how much of the explicit (γ) and the implicit (β) relations flow into the embeddings. Starting from their proposed parameter settings, we tune γ and β and conclude that a higher γ returns better performance for link prediction. This implies that the explicit links are more important than the implicit links. Due to its high computational cost, we compute 10 runs for the evolution-oriented approach and 5 runs for the noise-oriented approach.

A detailed overview of the applied embedding methods and the corresponding hyperparameters can be found in Table A.1

5

Visual Analysis with PCA

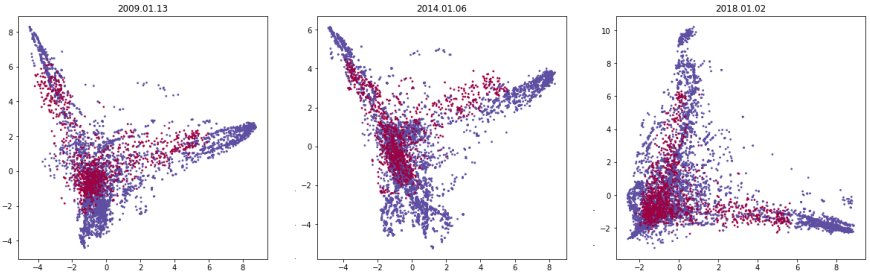
In this chapter, we provide an initial visual comparison of the different embedding methods across the evolution. This step is usually performed to examine how well the embeddings can describe the underlying data [3, 11, 28]. We utilize the Principal Component Analysis (PCA) [24] to map the embeddings of drugs and diseases into a two-dimensional space. We choose PCA over t-SNE [17] and UMAP [21], because the generation process of the projections is deterministic. Also, PCA highlights the global structure of the embedding space rather than distorting it [29]. We apply this technique on three different versions from the ontology evolution and also run it for the noise-oriented approach. Our initial assumption is that drug and disease embeddings form two clusters and are visually distinctive. Furthermore, we expect the layout to stay more or less the same across the evolution and despite the noise addition.

5.1 Evolution-oriented perspective

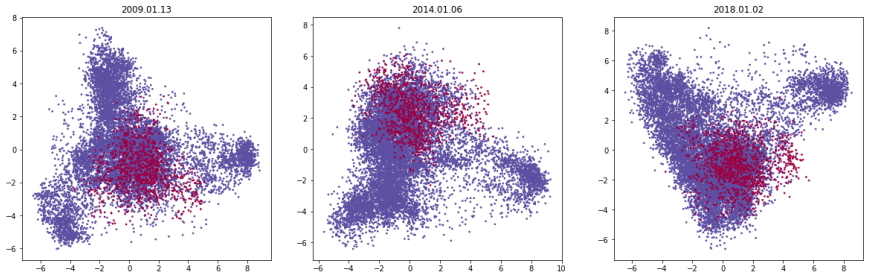
We select three versions (2009.01.13, 2014.01.06, and 2018.01.02) of the same seed and compare their representations. Figure 5.1 depicts the representation of the embeddings for each embedding method, where drug points are displayed in purple, and disease points in red. In 5.1a, we observe that the drug and disease clusters cannot be fully distinguished from each other as they overlap. Also, we notice that the diseases keep their shape, whereas the drugs are scattered more as the ontology evolves. Nevertheless, the structure of diseases being surrounded by drugs in a triangular shape is preserved. The explained variance ratio of the principal components is in total 13%, which means that PCA is unable to retain most of the embedding information on two dimensions.

In Figure 5.1b, the two-dimensional representation of the LINE embeddings are presented. The disease points stay close together and are surrounded by the drug points; however, there is no clear shape visible as the ontology evolves. In addition, the explained variance ratio of the principal components adds up to 15%, which is still very low. Therefore, we refrain from making any further statement.

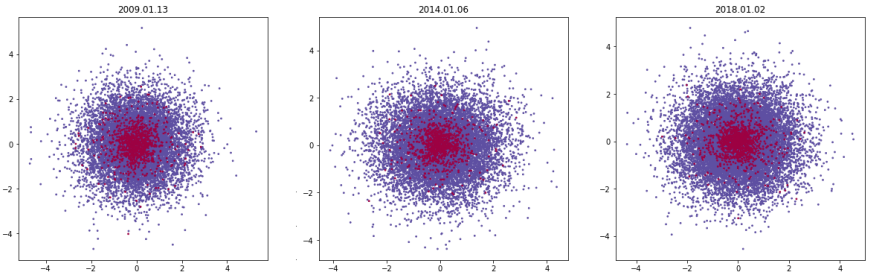
The BiNE representations are displayed in Figure 5.1c. The overall circular shape remains unchanged throughout evolution. Here again, the disease points stay dense and are surrounded by the drug points. The explained variance ratio makes up less than 2% so that the drug points and disease points contain only little information about the initial embeddings.



(a) GraRep



(b) LINE



(c) BiNE

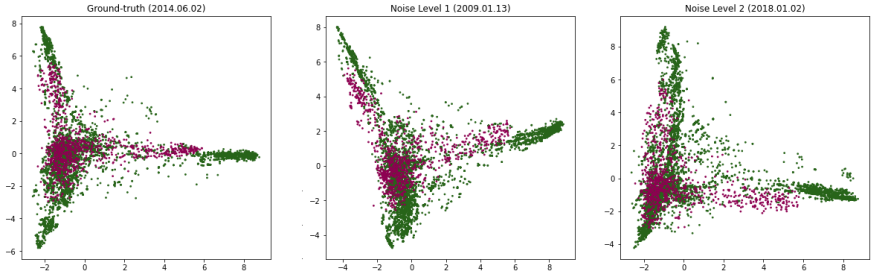
Figure 5.1: PCA projections on the evolution (seed=5). Drugs are purple and Diseases are red.

5.2 Noise-oriented perspective

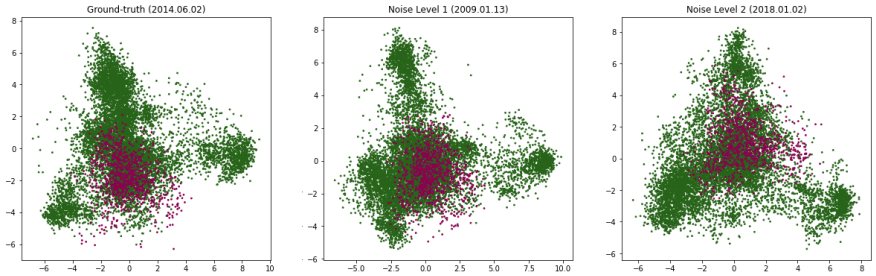
In the noise experiment, we apply PCA on the ground-truth and the two synthetic versions. Figure 5.2 presents the projections with different embedding methods, where drug points are displayed in green, and diseases in red. We notice that the overall structure is preserved despite the noise addition. However, noise level 1 is more similar in shape to the ground-truth than to noise level 2. This can be nicely observed for GraRep and LINE. Similar to the evolution-oriented perspective, we receive an explained variance ratio of 13% for GraRep, 15% for LINE, and lastly, 2% for BiNE. However, this time we notice that the shapes are preserved for all embedding methods despite the noise addition. Especially for LINE, we now observe a consistent pattern between the ground-truth version and the synthetic versions. In all projections, the diseases are concentrated in the middle, whereas the drug points are scattered.

5.3 Summary

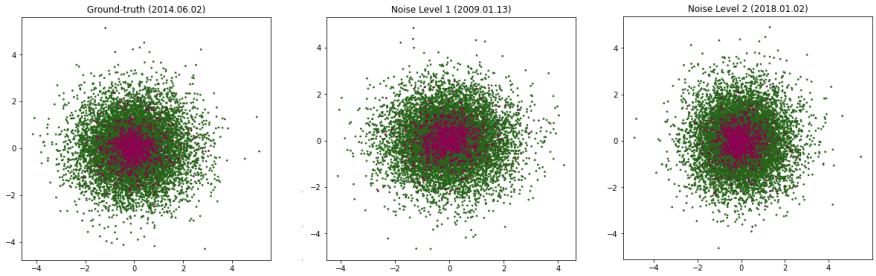
To conclude, we have observed several overlaps in the projections of the drugs and diseases. However, the drug points respectively the disease points are clustered together and region bound, thus revealing that the principal components can discriminate the two categories to a certain point. Nonetheless, PCA is unable to retain most of the information (variance) from the initial embeddings for any of the embedding methods. Out of the three methods, the GraRep embeddings return the best result with an explained variance ratio of 13% and a maintained layout throughout the evolution. Although the embeddings from LINE preserve more of the information (15%), the shape across the evolution is not explainable. In contrast, the BiNE embeddings retain the same shape; however, they keep almost no information about the original embeddings, which is surprising, as we expected BiNE to perform better and project two distinct clusters. However, we can only assume that our embeddings cannot be fully linearly described with PCA, making it a non-suitable method to project the DDA network. In Appendix A.3, we present the visualizations of the embeddings with the dimension reduction techniques t-SNE and UMAP.



(a) GraRep



(b) LINE



(c) BiNE

Figure 5.2: PCA projections on the noise experiment (seed=1). Drugs are green and Diseases are red.

6

Local Neighborhood Similarity

While embeddings may hold important structural and semantic information of a node, they are simply vectors in a multi-dimensional space. Therefore, we can measure the distance between these vectors with common metrics such as the cosine or euclidean distance. We utilize the local neighborhood (LN) similarity that is widely used to compare embeddings [2, 15, 25]. To this end, we define several distance (percentile) values for the neighborhood of two embeddings and report their similarity. The comparison is performed on embeddings of the same version, comparing each one with the remaining embeddings from different runs (10 out of 50 randomly selected runs). We do not use a cross-version comparison, because each version has a different set of nodes. This makes the comparison challenging, as we would have to first remove all the embeddings (nodes) that do not appear in each of the two versions to be compared. Furthermore, the existence about the removed nodes is still included in the remaining embeddings. The results would be distorted, and we would not be able to distinguish if it is due to the newly added edges or the inclusion of the removed nodes. Therefore, we conduct all comparisons using embeddings of the same version. In this task, we will answer the following two questions:

RQ1.1. Which distance (percentile) value reports the highest neighborhood similarity across the evolution?

RQ1.2. Can the best performing distance (percentile) value for each embedding method ensure a consistent neighborhood similarity across the evolution?

6.1 Approach

First, we calculate the euclidean and cosine distances between the embeddings of the same version with different seeds. The euclidean distance can be defined as the relative distance between two vectors from the zero point. Hence, if two vectors have a small euclidean distance, it implies that they are close to each other in the multidimensional space (i.e., in the same region) and thus similar in magnitude. In contrast, the cosine distance is computed from the angle between two vectors. Vectors that point in the same direction have a small cosine distance regardless of their magnitudes. In other words,

the cosine distance would consider those embeddings that have a similar structure to be close, even though they may be very far apart in the vector space. Therefore, it is often used when comparing documents of varied lengths and containing different term frequencies.

For both distance measures, we obtain a distance matrix for each version instance and set a threshold for the neighborhood such that only embeddings with a distance of less than r are included. The parameter r is defined by the percentiles of each distance distribution. We use the following percentile values: $radius = [0.05, 0.1, 1, 10, 20]$, where $r \in radius$. With two sets of neighborhoods for each entity, we use the Jaccard index and the overlap coefficient to evaluate their similarity. The Jaccard index is a widely used metric that, given two sets A and B, calculates the intersection of A and B over the union of A and B. In contrast, the overlap coefficient can be computed by the intersection of A and B over the size of the smaller set between A and B, as shown in Equation 6.1 and 6.2.

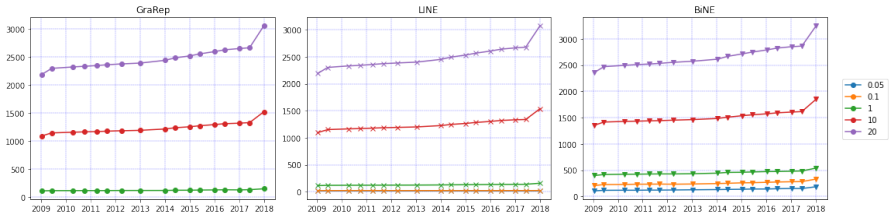
$$Jaccard = \frac{\mid A \cap B \mid}{\mid A \cup B \mid} \tag{6.1}$$

$$Overlap = \frac{\mid A \cap B \mid}{\min(\mid A \mid, \mid B \mid)} \tag{6.2}$$

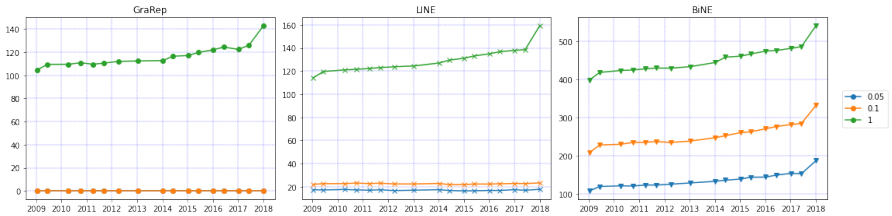
The overlap coefficient is a lesser known method, but its benefit is that it indicates how much of the smaller set is contained within the larger. Therefore, it represents the difference between the two sets. Since the embedding methods determine when embeddings are close to each other, we prioritize the euclidean rather than the cosine distance for the evaluation. All the embedding methods that we use focus on the proximity of the nodes. To give an example, LINE uses the first and second proximity when generating embeddings, thus nodes sharing the same neighborhood have similar embeddings. For these reasons, we put less weight on the cosine distance. Nevertheless, we present both distance measures, as our embeddings consist of 100 dimensions, where the vector space is bound to be sparse.

6.2 Evolution-oriented perspective

Prior to reporting the similarity measures, we first present the neighborhood size taking into account the embedding method and the percentiles. Figure 6.1 shows the average size of the neighborhood computed with the euclidean distance across the evolution. We notice that, with a larger percentile, the neighborhood sizes of the different embedding methods become more similar. Only for $r \leq 1$ do we observe different neighborhood sizes. GraRep and LINE show no great variability with lower values of r. However, with BiNE, we notice that around 200 neighbors are included at $r = 0.1$, whereas twice as many are captured with $r = 1$. Nonetheless, we can compare the embedding methods more easily with a larger percentile value, as the similarity measures can be seen as equivalent at this point. All the graphs depict an increase in neighborhood size as



(a) Higher r' values



(b) Lower r values

Figure 6.1: Neighborhood sizes for GraRep, LINE and BiNE

the ontology evolves. This is expected, as the ontology becomes populated with new nodes and edges, making the vector space denser. The neighborhood sizes for the cosine distance are almost identical. The same figure can be therefore found in Appendix A.4.

Moving to the actual results of the neighborhood similarity, we present the values in Figure 6.2, where each point in a plot represents a DDA version. The similarity metrics of the embedding methods turn out be very different. Figure 6.2a presents the Jaccard index and the overlap coefficient with the euclidean distance. The same similarity metrics can be found in Figure 6.2b, which depicts the results with the cosine distance. We initially notice no major differences between the results of the euclidean and the cosine distance. In a few cases, we see a slightly lower similarity value for the cosine distance (e.g., GraRep with $r = 1$). However, this minor difference can be neglected, since the overall pattern of the results across the evolution is preserved. Therefore, we focus on the similarity measures of the euclidean distance for the rest of this chapter. We first notice that the values of the overlap coefficient are higher than the Jaccard index. This is expected, since the denominator in the overlap coefficient is smaller than that of the Jaccard index. Nonetheless, for GraRep and LINE, both similarity metrics show an almost identical pattern in accordance with the different percentile values. In contrast, with BiNE we observe a slightly different behavior, where, for example, the Jaccard index is proportionally higher than the overlap coefficient at $r = 0.05$. However, we refrain from examining this difference, since our focus is on the Jaccard index for the rest of this chapter. We leave this exploration to future works. The terms *similarity* and *Jaccard index* are henceforth used interchangeably.

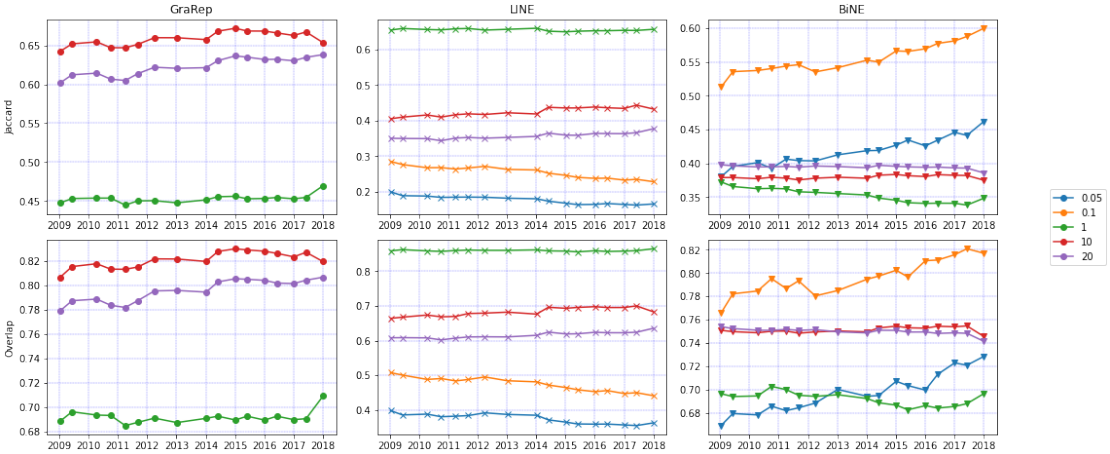
Considering the Jaccard index, we can see no curve for GraRep at $r = 0.05$ and $r = 0.1$. This occurs because it fails to return valid neighbors at a low percentile value. With $r \geq 1$, the Jaccard index curves are visible in the plot, and at $r = 10$, the highest similarity value is reached at approximately 0.65. At this point, the neighborhood size

consists of around 1'000 neighbors. The lowest similarity value of 0.45 is reached at $r = 1$, where the neighborhood size accounts for around 100 neighbors. From this, we can deduce that a higher percentile and thus a larger neighborhood returns better similarities for GraRep. With LINE, the highest Jaccard index is reached at $r = 1$, with around 100 neighbors. Anything below or above this reports a worse similarity value; however, we notice that the similarity for higher r values differs less than those for lower ones. Although the distance between the higher r values and $r = 1$ is much larger, the Jaccard index deviates only around 0.2 from the highest similarity value. Therefore, we can make the same conclusion as for GraRep, where higher percentiles return better similarities. Finally, in the case of BiNE, the highest similarity is reached with $r = 0.1$, which is the lowest percentile value out of the three embedding methods. The neighborhood size at this point increases from 200 to 300 along the evolution. At the same time, the Jaccard index demonstrates an evident increase of 0.1 across the evolution. This means that the neighborhood similarity increases across the evolution with a steadily growing neighborhood. We do not observe this evident behavior with the other embedding methods. Although a slight increase along the evolution is visible in some cases, the increase is not as apparent as in BiNE. In addition, higher or lower r values show no distinct differences; thus, we can consider them to be equally poor. To summarize, for GraRep and LINE, higher r values report better similarity as opposed to BiNE, which returns a high similarity for a lower r. In general, the minor fluctuations in the plots can be explained by the increase in neighborhood size as the ontology evolves.

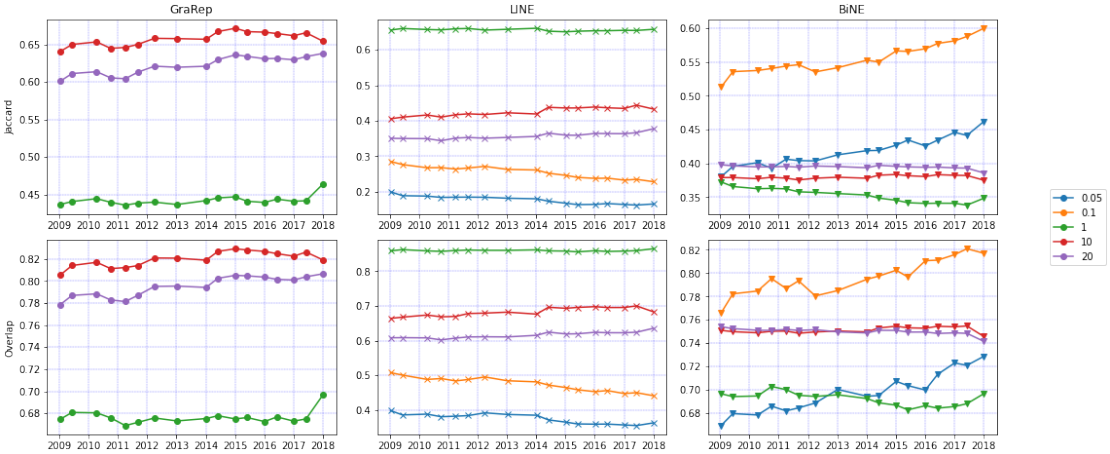
To provide a conclusive picture of the above interpretations, Figure 6.3 presents the average Jaccard index and neighborhood size computed from the different versions. In Figure 6.3a, each embedding method reaches its peak at a different r value. GraRep reports the highest similarity at $r = 10$, whereas the other two embedding methods hold their peaks at a much lower value of r. LINE shows the highest similarity at $r = 1$ whereas BiNE depicts a slightly lower percentile at $r = 0.1$. Figure 6.3b demonstrates that GraRep and LINE have an almost identical increase in neighborhood size across percentile values. BiNE, on the other hand, reports larger neighborhood sizes for each percentile. However, at higher r values, the neighborhood size of BiNE becomes closer to that of GraRep and LINE. From these findings, we can derive that the BiNE embeddings are slightly denser compared to the GraRep and LINE ones.

In order to analyze and interpret the respective distance values, we utilize *Welch's t-test* which is an adaption of the *unpaired t-test*, whereby an equal variance is usually a prerequisite. Applying *Levene's test* proved that the variances in several samples are not equal for GraRep, LINE and BiNE (see Appendix A.4.1). We proceed with *Welch's t-test* and compare the mean of the Jaccard indexes from each version to every other version. As we are running multiple hypothesis tests, we use the Bonferroni method to correct the p values. We refrain from reporting the results of both distance metrics since the similarity measures of the euclidean and cosine distance are almost identical. Therefore we present the *Welch's t-test* results along with the euclidean distance in Figure 6.4. The results with the cosine distance can be found in the Appendix A.4.1.

For GraRep, most of the versions return no significant difference in the mean at $r = 1$ with a few exceptions that predominantly reject the null hypothesis (2011.04.04

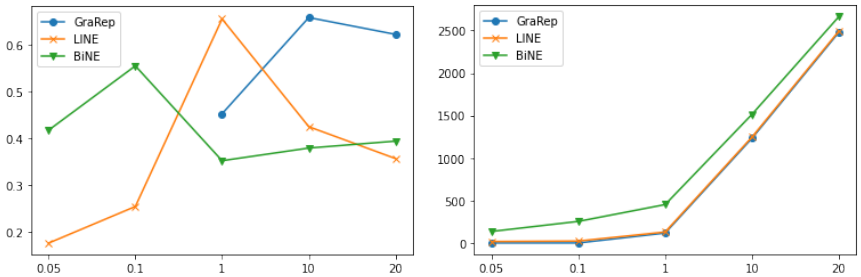


(a) Euclidean distance



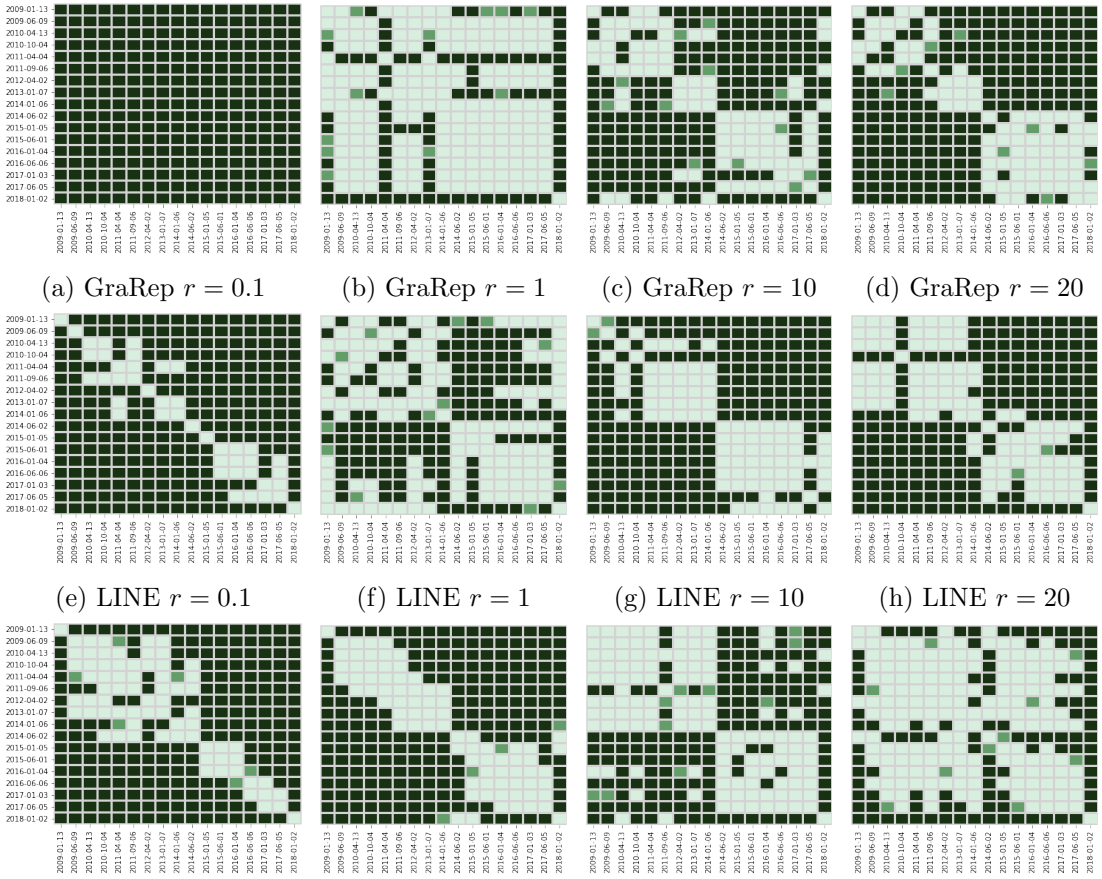
(b) Cosine distance

Figure 6.2: Similarity metrics for GraRep, LINE and BiNE



(a) Average Jaccard Index (b) Average neighborhood size

Figure 6.3: Averages across the evolution for different percentile values (x-axis) with euclidean distance



(a) GraRep $r = 0.1$ (b) GraRep $r = 1$ (c) GraRep $r = 10$ (d) GraRep $r = 20$

(e) LINE $r = 0.1$ (f) LINE $r = 1$ (g) LINE $r = 10$ (h) LINE $r = 20$

(i) BiNE $r = 0.1$ (j) BiNE $r = 1$ (k) BiNE $r = 10$ (l) BiNE $r = 20$

Figure 6.4: Welch's t-test for LN similarity comparison with $\alpha = .05$ (dark green stands for rejected null hypothesis and light green for accepted null hypothesis)

and 2018.01.02). For the version at 2018.01.02, we are aware, that at this point, the largest changes occurred in the ontology; this can also be observed in Figure 6.2. In the same figure (6.2), we notice a slight decrease at 2011.04.04 that later recovers in the consecutive versions. This indicates a drop in the similarities, which explains the high number of rejected null hypotheses at this point. For $r = 10$ and $r = 20$, the results show a very similar pattern. Neighboring versions demonstrate no significant difference in the mean, whereas farther versions reject the null hypothesis. Therefore, we can conclude that at $r = 1$, GraRep outputs the most consistent Jaccard indexes across the evolution. An exception is at $r = 0.1$, where we have previously seen that no valid neighbors were selected, making a comparison impossible. Furthermore, the higher the r value, the more inconsistent the results, especially for those versions that are farther apart. For LINE, we notice no profound differences across different r values. At $r = 0.1$, we can observe the most restricted behavior, whereby only a few neighboring versions accept the null hypothesis. For higher r values, this restriction dissolves. At $r = 1$ in particular, we find that several versions demonstrate no significant differences in the mean. Looking at version 2014.06.02, we notice that all the prior versions report a significant difference, whereas the consecutive ones prove to be non-significant. This constitutes the turning point from which the ontology introduces a breaking change that shifts the mean such that it shows significance. In general, we observe once more that neighboring versions are more likely to accept the null hypotheses than those that are farther apart. With BiNE, we notice a likely opposite behavior to GraRep. At $r = 0.1$ and $r = 1$, only the closest neighboring versions accept the null hypothesis; however, with a higher r, the Jaccard indexes start to converge along the evolution, proving an evident decrease in significant differences. Nonetheless, we observe the turning point at 2014.06.02 again, and most of the versions report a significant difference from version 2018.01.02.

6.3 Noise-oriented perspective

The same neighborhood similarity algorithm is applied to the ground-truth (GT) version (2014.06.02) and the respective synthetic versions. In this perspective, we compare the ground-truth with each synthetic version that comprises a different noise level. In addition, we compare the ground-truth version with different seeds referred to as G'. This provides an indication of how the noise addition affects the original Jaccard index. Table 6.1 presents the similarity metrics computed with the euclidean distance. The results with the cosine distance can be found in Appendix A.4.2.

Similar to the evolution-oriented perspective, GraRep only provides values with $r \geq 1$. From there, we observe that noise level 1 reports slightly higher values than noise level 2. Additionally, compared to GT, the difference for noise level 2 is larger than for 1, even though the neighborhood size does not seem to differ greatly. Observing the Jaccard index over the different percentile values, we notice that the distance between GT and the different noise levels remain more or less constant. For the LINE embeddings, the Jaccard indexes for both noise levels show a marginal difference to GT for each percentile value. Moreover, among the noise levels, the Jaccard index deviates by at most 0.04. Here as

	$euc_{<0.05\%}$	$euc_{<0.1\%}$	$euc_{<1\%}$	$euc_{<10\%}$	$euc_{<20\%}$
GraRep					
GT - GT'	-	-	0.46 [104]	**0.67** [1'075]	0.63 [2'161]
GT - Noise L1	-	-	0.42 [104]	**0.60** [1'074]	0.56 [2'161]
GT - Noise L2	-	-	0.40 [104]	**0.57** [1'076]	0.53 [2'161]
LINE					
GT - GT'	0.18 [16]	0.27 [21]	**0.65** [113]	0.43 [1'085]	0.36 [2'171]
GT - Noise L1	0.17 [16]	0.27 [21]	**0.62** [113]	0.41 [1'085]	0.35 [2'171]
GT - Noise L2	0.17 [15]	0.26 [20]	**0.58** [113]	0.40 [1'085]	0.34 [2'170]
BiNE					
GT - GT'	0.38 [106]	**0.50** [203]	0.37 [410]	0.38 [1'331]	0.40 [2'333]
GT - Noise L1	0.36 [107]	**0.48** [207]	0.36 [408]	0.37 [1'335]	0.39 [2'333]
GT - Noise L2	0.35 [108]	**0.46** [204]	0.36 [422]	0.38 [1'336]	0.39 [2'331]

Table 6.1: Jaccard index for GraRep, LINE and BiNE with euclidean distance. Number in brackets $[\cdot]$ is the neighborhood size.

well, the neighborhood size does not change much despite the different noise levels. The BiNE embeddings behave similarly, and no major differences can be observed. In some cases, the two noise levels report the same similarities, although it is not the exact same value due to rounding differences. The Jaccard indexes between the noise levels deviate by at most 0.02, which is the lowest among the embedding methods. Especially with $r \geq 1$, the values are almost identical, although the neighborhood sizes are not equal. Overall, it is surprising how close the Jaccard indexes of the different noise levels are to GT. Considering that we added five times more edges in level 2 than in 1, the difference can be seen as minor. The marginal differences in the neighborhood size among the levels are also unexpected. Here again, we can see that the neighborhood sizes for GraRep and LINE are similar along the percentile values. However, BiNE at $r = 0.05$ starts off with a much larger neighborhood size but the difference diminishes with higher percentiles.

For further analysis, we turn to the evolution-oriented perspective, where we first apply *Levene's test* (see Appendix A.4.2) followed by the *Welch's t-test* to find out if there is a significant difference in the mean between the Jaccard indexes of the ground-truth and the respective synthetic versions. We apply the statistical tests to four different r values and present the results in Table 6.2.

Surprisingly, all the comparisons report a significant difference in the mean along the different percentile values. Observing the t-statistic, we can see a consistent behavior for GraRep and LINE, whereby noise level 1 reports a lower value than 2 across the board. With an increasing r, we notice that the difference in the t-statistic increases much more than for LINE. In contrast, BiNE shows the opposite behavior with $r \geq 1$, whereby the t-statistic for noise level 2 is lower than for 1. Moreover, noise level 2 displays a value twice that of noise level 1. This is very unexpected and indicates that the mean of noise level 2 is closer to the ground-truth than that of noise level 1.

	GraRep		LINE		BiNE	
	statistic	p-value	statistic	p-value	statistic	p-value
$cos_{<0.1\%}$						
GT − Noise L1	-	-	7.7236	<.0001	6.5710	<.0001
GT − Noise L2	-	-	15.6866	<.0001	13.0532	<.0001
$cos_{<1\%}$						
GT − Noise L1	29.0419	<.0001	31.6608	<.0001	5.7017	<.0001
GT − Noise L2	42.0600	<.0001	77.9198	<.0001	3.9185	0.0010
$cos_{<10\%}$						
GT − Noise L1	54.7415	<.0001	10.7561	<.0001	10.7527	<.0001
GT − Noise L2	81.2714	<.0001	16.6438	<.0001	5.5429	<.0001
$cos_{<20\%}$						
GT − Noise L1	53.4194	<.0001	5.8066	<.0001	33.4405	<.0001
GT − Noise L2	89.0791	<.0001	11.4771	<.0001	16.0558	<.0001

Table 6.2: Welch's t-test for the noise experiment with the euclidean distance

6.4 Discussion

In the evolution-oriented perspective, we found that the Jaccard index for neighboring versions shows no significant difference in most cases. This was proven regardless of whether we computed it with the euclidean or the cosine distance. Hence, we can infer that the near-term evolution of an ontology has no significant effect on the neighborhood similarity. Apart from this, we noticed that farther located versions reject the null hypothesis in several cases. However, this is not guaranteed, as we also observed patterns where a noticeable change in the ontology leads to a shift in the mean such that all the previous versions show a significant difference to the subsequent ones. In addition, taking the different percentile values into consideration, none of the embedding methods provide a clear consistency over different r values.

In the noise-oriented perspective, both noise levels reported a significant difference in the mean across percentile values. We also noticed that the difference in the t-statistic between the noise levels is larger for GraRep than for LINE, which implies that the addition of noise affects the former to a greater extent. In the case of BiNE, we found that noise level 2 reports a lower t value than level 1 with $r \geq 1$. This is very unexpected and will be discussed in future works.

In both perspectives, we received consistent results regarding ***RQ1.1.*** BiNE reports the best performing percentile value at $r = 0.1$, LINE at $r = 1$ and lastly GraRep at $r = 10$. This was observed from the Jaccard index of both distance metrics, euclidean and cosine. Regarding ***RQ1.2.***, we took the best percentile values and analyzed the results of *Welch's t-test* across the evolution. None of the embedding methods fully show a consistent pattern across the evolution at their best performing r. It turns out that several parameters in the task influence the similarities and consequently the number of significant differences in the mean. Nevertheless, when focusing on the best performing

distance value, we can clearly see that LINE rejects the fewest null hypotheses out of the three embedding methods. GraRep and BiNE perform the worst in terms of consistency at their best performing r values. Furthermore, across the different r values, LINE is the closest to demonstrating a common pattern.

To conclude, the neighborhood similarity task shows that we can determine a percentile value that clearly indicates the highest Jaccard index. However, it should not be assumed that the best performing percentile value also outputs the highest consistency across the evolution. This is highly dependent on the embedding method, the percentile value, the distribution of the embeddings in the vector space and the severity of the changes across the ontology evolution. Consequently, there is also no guarantee that embeddings will report consistent results across different percentiles. Although this was anticipated, we did not expect such drastic pattern changes with different r values. Nonetheless, dealing with parameter adjustments in the task tells us how sensitively the embeddings respond to such events. This provides further evidence of the robustness of an embedding. In addition, greater changes in the ontology very likely shift the mean of the Jaccard indexes and thus report a significant difference. This is understandable, as the addition or removal of multiple nodes is bound to change the neighborhood. However, the neighboring versions are usually not concerned, as most of them do not comprise major differences and therefore report no significance in the mean.

7

Link Prediction

Link Prediction allows to find new relationships between entities, and it has been extensively used as a down-stream task to evaluate embeddings [8, 32]. Similarly to Hasan et al. [16], we define it as a supervised learning problem. The goal of the prediction model is to differentiate between positive and negative links, which requires training data of both types. However, as mentioned in the previous chapter, links are only predicted for known nodes with respectively generated embeddings. Also, while we have access to validated positive samples, in most cases, validated negative samples are unfortunately not provided. Those are often randomly selected from not-linked edges, which enhances the risk of adding unnecessary noise, consequently acting counterproductive. The incorporated link prediction from the BioNEV package does precisely this. The negative samples are randomly selected with the same number of positive edges. Furthermore, the edge selection is random on all existing nodes, allowing negative edges that include drug-drug or disease-disease links. Such relations are not allowed in a bipartite graph. Given these implications, we will answer the following two questions:

RQ2.1 How severely is the link prediction performance across the evolution affected when one modifies the prediction algorithm?

RQ2.2 How does the contribution of domain-specific restrictions influence the link prediction performance across the evolution?

7.1 Logical Restriction

We extend the link prediction from BioNEV with logical restrictions by (1) allowing only drug-disease links for negative samples (2) providing a reduced selection of reliable negative edges. The second restriction is inspired by Wu and Liu [31] - they introduce a method that is called *Reliable Negative Samples Selection*. Their algorithm is based on the widely used assumption that similar drugs may treat similar diseases. The idea is simple: nodes with an (indirect) relation to each other are not included in the selection. How far this boundary lies can be defined by k, which represents the number of steps or rather the path length between two nodes. As a result, the *k-step* commuting matrix captures the existence of a path between two nodes.

Wu and Liu [31] use a *3-step* algorithm in their research; however, the graph they are working with is much smaller compared to the DDA network. Therefore, we apply the *3-step* as well as the *7-step* method in our work. An example for a *3-step* algorithm is presented in Equation 7.1, 7.2 and 7.3. The first equation (7.1) includes all *1-step* neighbors, where A_{ds} is simply the drug-disease association matrix:

$$D1 = A_{ds} \tag{7.1}$$

Equation 7.2 contains the *3-step* neighbors, such that all paths along $Drug-Disease-Drug-Disease$ are captured.

$$D3 = A_{ds} \times A_{ds}^{T} \times A_{ds} \tag{7.2}$$

Finally, all the commuting matrices are summed up as in Equation 7.3. The *2-step* commuting matrix can be omitted as these paths are already included in $D3$.

$$D = D1 + D3 \tag{7.3}$$

Now with D, we can examine, if there exists any path up to length 3 between two nodes. Nodes with no path in between them have the value zero. Therefore, when selecting the negative edges, we first check whether the value between the proposed nodes is 0. If yes, we include it in the negative samples from the selection. Apart from this, we use the same link prediction settings as provided in the BioNEV package that consists of a Logistic Regression binary classifier with an 80% training and 20% test set.

7.1.1 Evolution-oriented perspective

In Figure 7.1, we report the link prediction performance across the evolution where column-wise: 1) no reliable negative samples exist, 2) *3-step* neighbors are excluded from the negative sample selection, and 3) *7-step* neighbors are excluded. The performance for 1) and 2) show no substantial difference at first glance with a shift of around 0.01. In contrast, the results with the *7-steps* method demonstrate a major increase in performance with an AUC ROC of approximately 0.96. Additionally, the results become more distinct with an increase in the step size. This can be clearly seen at 2018.01.02 (last version), where the gap to the other values increases significantly. Comparing the embedding methods, we observe that the LINE embeddings show slightly poorer performance for the *0-step* and *3-step* method than the GraRep embeddings; however, the results of LINE turn out higher in the *7-step* method. BiNE reports similar results to GraRep except for the *7-step* method, where it is lower by approximately 0.03. The AUC ROC for GraRep, LINE, and BiNE show minor fluctuations across the evolution, and only in 2018.01.02, there is a significant increase visible. This can be explained by our findings in Chapter 3, where we learnt that most of the nodes and edges were added at this point. Besides that, the fluctuation pattern of different step sizes remains the same for all embedding methods, which reveals that the prediction model stays unchanged for all versions and is not affected by the restricted negative sample method.

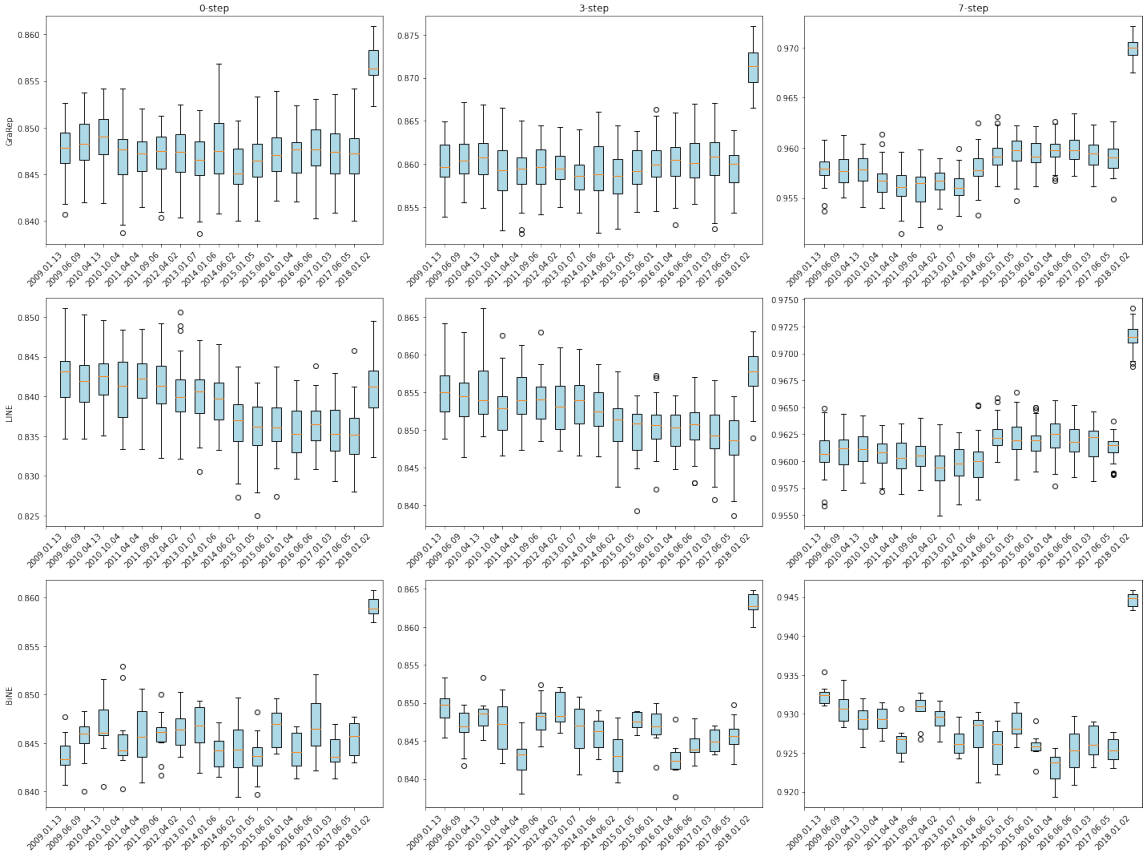
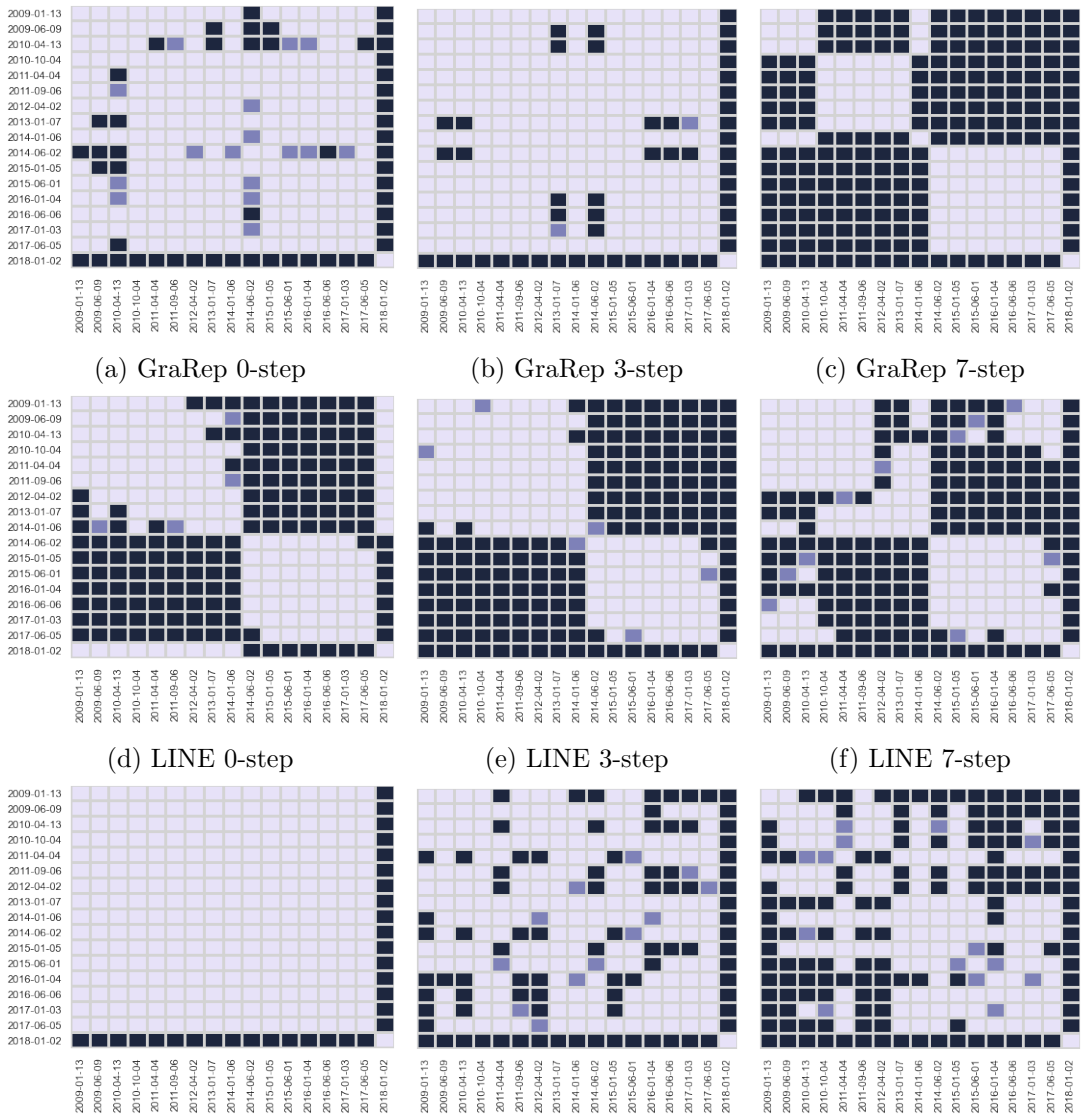


Figure 7.1: Link prediction performances (AUC ROC)

We additionally apply *Welch's t-test* to examine if the mean of the AUC ROC values across versions differs significantly. By running *Levene's test*, we find that the variances in several samples are not equal for GraRep, LINE and BiNE (see Appendix A.5.1). Hence, we proceed with *Welch's t-test* that does not require an equal variance but can still be used to compare the means between two groups. We run the test for each version comparing it to the remaining ones. Since we are testing multiple hypotheses, we again use the Bonferroni method to correct the p values. The adjusted p values are presented in Figure 7.2.

We observe that the results demonstrate very different patterns for GraRep, LINE, and BiNE. In the *0-step* and *3-step* method, GraRep shows almost no significant differences between the versions. BiNE reports no significance in the *0-step* method (with one exception), but this changes drastically with a larger step size. LINE differs considerably, whereby the first half of the versions demonstrate no significance between the means, but the other half rejects the null hypothesis. The version at 2014.06.02 acts as a turning point from where the exact opposite behavior can be observed. All consecutive versions report no significant difference, whereas the versions before 2014.06.02 reject the null hypothesis. This indicates that there are two groups with significantly different



(a) GraRep 0-step (b) GraRep 3-step (c) GraRep 7-step

(d) LINE 0-step (e) LINE 3-step (f) LINE 7-step

(g) BiNE 0-step (h) BiNE 3-step (i) BiNE 7-step

Figure 7.2: Welch's t-test for Link prediction performance with $\alpha = .05$ (dark blue stands for rejected null hypothesis and light blue for accepted null hypothesis)

means divided at the turning point. When comparing the GraRep and LINE results of the *7-step* method, the patterns resemble each other. The turning point remains at 2014.06.02 for both methods, with a few exceptions that reject the null hypothesis. For all embedding methods, the last version (2018.01.03) remains an exception because it behaves differently from the remaining versions. It rejects the null hypothesis regardless of the other version's proximity, and only in the *0-step* results of LINE, does it show a similar mean with the first half of the versions.

7.1.2 Noise-oriented perspective

We apply the same prediction algorithm on the ground-truth (GT) and the synthetic versions with different step sizes. Table 7.1 presents the AUC ROC for each step method. Overall, there is only a small margin between the values of the ground-truth and those of the synthetic versions. Nonetheless, the results at noise level 2 show a more considerable difference compared to noise level 1. We also notice that the difference increases with a larger step size. As an initial assessment, we can state that the results remain stable despite the noise addition.

	0-step	3-step	7-step
GraRep			
GT	84.54	85.80	95.91
Noise L1	84.46	85.84	95.72
Noise L2	84.91	86.52	97.32
LINE			
GT	83.71	85.15	96.30
Noise L1	83.85	85.22	96.05
Noise L2	83.75	85.58	97.57
BiNE			
GT	84.46	84.49	92.92
Noise L1	84.46	84.62	92.86
Noise L2	84.35	84.74	94.27

Table 7.1: Link prediction performance (AUC ROC in %) for the noise experiment

Similar to the evolution-oriented perspective, we first run *Levene's test* followed by *Welch's t-test*. Here, we compare the ground-truth version with the synthetic versions for each step method, so that we can examine if the noise addition demonstrates a significant effect on the link prediction performance. The results of *Levene's test* (see Appendix A.5.2) verify that the variance for the ground-truth and each synthetic version is the same across all the embedding methods; thus, the null hypothesis can be accepted. However, for consistency reasons, we proceed with *Welch's t-test* and apply it to the ground-truth and the synthetic versions. Table 7.2 presents the results. We notice that, for GraRep, the results of the *0-step* and *3-step* methods only show a significant difference at noise level 2. For the *7-step* method, both noise levels show a significance

in the mean compared to the ground-truth. Regarding the *0-step* method of LINE, both noise levels report p values above 0.05, which implies that the null hypothesis holds. Noise level 2 shows a significant difference only in the *3-step* method for LINE, and, both noise levels reject the null hypothesis with the *7-step* method. In contrast, BiNE reports a significant difference only in the *7-step* method at noise level 2. From these results, we can infer that the prediction performance of BiNE is the least affected by the noise addition, whereas GraRep demonstrates the largest changes in performance. In general, the difference between the p values of the two noise levels is much larger for GraRep than it is for LINE or BiNE.

	GraRep		**LINE**		**BiNE**	
	statistic	p-value	statistic	p-value	statistic	p-value
0-step						
GT − Noise L1	-0.4922	0.6285	-0.8073	0.4301	-0.6774	0.5205
GT − Noise L2	-2.4509	0.0267	-0.2389	0.8141	0.2517	0.8077
3-step						
GT − Noise L1	0.3657	0.7190	0.5238	0.6070	-0.6097	0.5606
GT − Noise L2	-6.8903	<.0001	-2.7339	0.0138	-1.0782	0.3125
7-step						
GT − Noise L1	-2.6821	0.0169	-3.3006	0.0058	0.3779	0.7203
GT − Noise L2	-24.4994	<.0001	-21.7326	<.0001	-9.3775	0.0003

Table 7.2: Welch's t-test (AUC ROC) for the noise experiment

7.1.3 Discussion

In the evolution-oriented perspective, we saw that the AUC ROC value for all embedding methods remains similar across the evolution with minor fluctuations. The last version (2018.01.02) is an exception, which reports a larger AUC ROC value for every step size. This is not surprising as we expect larger changes in the ontology to impact the performance of the prediction model. We initially stated that we consider the link prediction performance to determine the robustness. However, this is not trivial, as multiple factors can influence the performance of a prediction model. One of them is, that we run the prediction task with different versions respectively different states of knowledge. In addition, we are forced to include a randomness factor, as we do not have access to any validated negative samples. We try to minimize this instability with the logical restriction algorithm. By keeping these facts in mind, we compare the prediction performances and consequently the robustness of the embeddings. We notice that the number of significant differences between the versions increases with a larger step size. Therefore the robustness across versions gets compromised, whereas at the same time we receive a higher prediction performance by adding the logical restriction, This is an unexpected trade-off, because we initially believed that reducing the randomness in the negative selection process would output more consistent results. Only with LINE, we

observe the least harm where the pattern of different step sizes is more or less preserved.

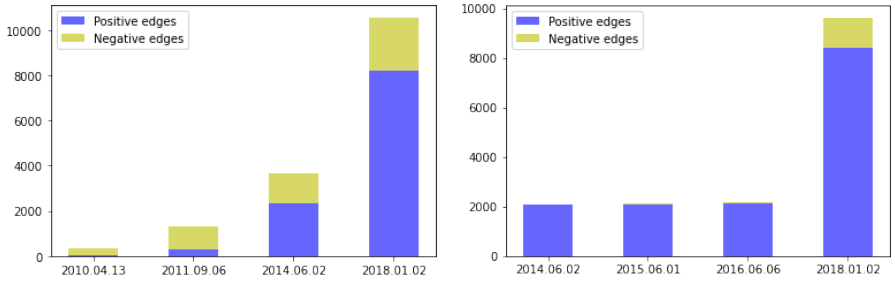
In the noise-oriented perspective, we found that the addition of noise does not affect the prediction results very heavily. Still, the performance values of BiNE are less affected by the noise than for LINE and GraRep. This is proven particularly by the *3-step* method for BiNE, where both noise levels accept the null hypothesis of an equal mean to the ground-truth. Unlike GraRep and LINE that reject the null hypothesis at noise level 2 for this step size.

In both perspectives, we observed that a larger step size increases the prediction results. Moreover, the variance becomes smaller, from which we can deduce that the model is able to make more precise predictions. The prediction results of the LINE embeddings have further shown a bigger performance boost when applying the logical restriction compared to GraRep and BiNE.

To answer ***RQ2.1***, we have proven that the results of the prediction task changes when adjusting the selection process of the negative sample. Using random selection not only adds unnecessary noise, but we are also unable to confidently present and interpret the results due to the randomness of the selection process. We do not know about the structure/distribution of the negative samples, which might lead to possible performance alternations that we are unable to explain. With the presented restriction algorithm, we are aware of the selection process, and we can ensure that it complies with the domain context. Regarding ***RQ2.2***, we have proven that the restricted negative sample selection improves the performance of the prediction model. In addition, the logical restriction leverages the model in making more precise predictions. However, in terms of robustness, GraRep and BiNE show a consistent pattern across the evolution only in the *0-step* and *3-step* method. In contrast, LINE generally divides the versions into two groups with different means; therefore, consistency is only guaranteed within the group. Nonetheless, when taking a different perspective of robustness so that we consider a consistent result over different step methods as robust, LINE clearly outperforms the other two methods.

7.2 Future Links

In this experiment, we train a model at a certain point in time and use future links in the test set for evaluating the prediction. The goal is to assess the performance of the prediction model for validated future links to find if the trained embeddings can predict the links correctly within their current state of knowledge. We train two models with the data available from 2009.01.13 and 2014.01.06, referenced as $V1$ and $V2$. The prediction model for this experiment is built as follows: the training set consists of the whole edgelist data available at t_0 where $t_0 \in [2009.01.13, 2014.01.06]$. The test set of the version at t_i includes the delta of t_0 and t_i, and only nodes that exist in both versions are covered. As the test set consists of validated positive and negative cases, it is very likely to become imbalanced. Taking this into account, we choose the appropriate evaluation metrics. Prior to the evaluation, we first look at the distribution of the positive P and negative N links at each t_i. There are two use cases, either $P >> N$ or $P << N$. The number of positive and negative edges at the corresponding t_i are presented in Figure 7.3. In



(a) 2009.01.13 (b) 2014.01.06

Figure 7.3: Number of positive and negative edges

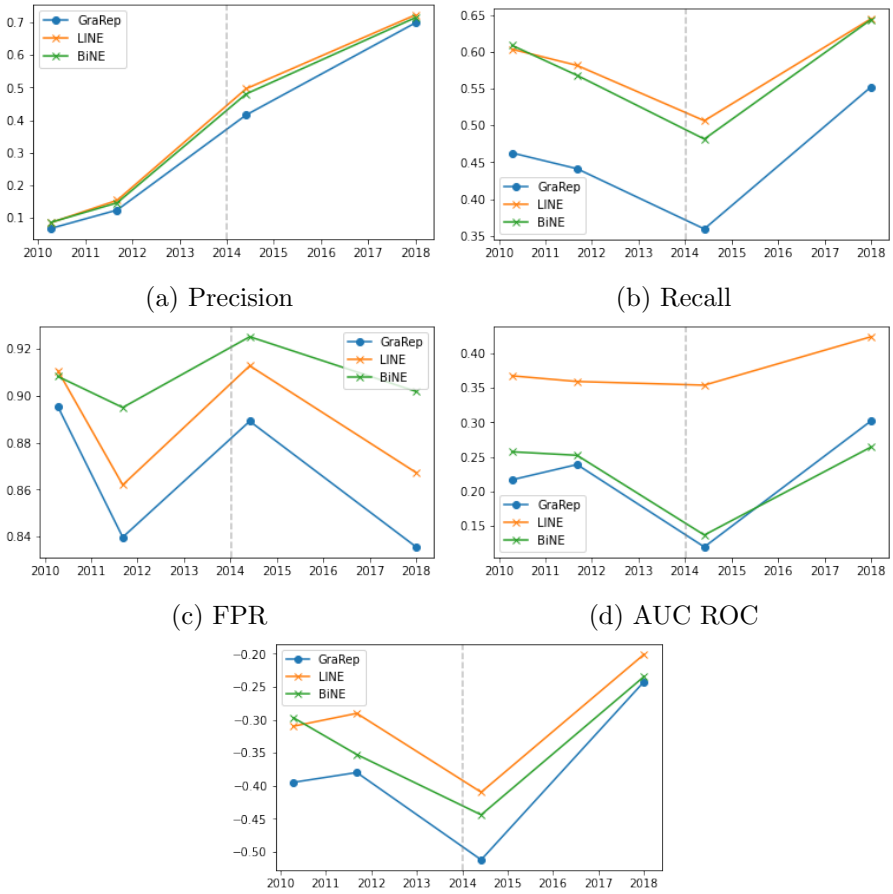
Figure 7.3a the test set in the first half consists predominantly of negative edges; however, this changes in the other half where there are mainly positive edges. Figure 7.3b shows that the positive edges dominate across all versions. Except for t_4 (last version), the negative edges are not clearly visible in the graph as they are located below 100. This information is essential, since it influences how we evaluate the prediction model in the following subsections.

7.2.1 State of knowledge at V1

Since both $P >> N$ and $P << N$ occur in this group of test set, we start by looking at the performance of the first half (t_1 and t_2) and then move on to the second half (t_3 and t_4). The first half contains more negative than positive edges; thus, we would intuitively state that the model predicts the true negatives correctly and a certain amount of the true positives as negative (false negative). Therefore, we focus on the positive cases and study the precision and recall, also known as the true-positive rate (TPR). In addition, we use another metric called *Matthew's correlation coefficient (MCC)* [20], also known as *phi-coefficient*, which takes into account all four values from the confusion matrix. MCC returns a value between -1 and +1, where -1 represents a negative correlation and +1 a perfect positive correlation. A high correlation value means that both classes, in our case the positive and the negative edges, are predicted well. If the correlation value is 0, it means that the prediction is similar to flipping a coin and in the case of a low correlation value, we would get an inverse prediction, where positive edges are predicted as negative and vice versa. Consequently, we aim for a high correlation value close to +1 for our prediction model. The formula is shown in Equation 7.4.

$$\text{MCC} = \frac{\text{TP} \times \text{TN} - \text{FP} \times \text{FN}}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}} \tag{7.4}$$

All the evaluation metrics are presented in Figure 7.4. The vertical dashed line marks the turning point where the number of positive edges becomes larger than the negative ones.



(a) Precision

(b) Recall

(c) FPR

(d) AUC ROC

(e) MCC

Figure 7.4: Performance metrics at 2009.01.13

As mentioned above, we first focus on Figure 7.4a and 7.4b. The first half reports a very low precision of approximately 0.1. This can be explained by the fact that the model predicts a high number of false positives, thus labeling part of the negative edges as positive. The recall values remain around 0.4 and 0.6, implying that the model predicts roughly 50% of the true positives as negative. Since the number of negative edges is disproportionally higher than the positive ones, we expect the model to predict more negative cases. In comparison, the precision is lower than the recall in the first half; we can therefore conclude that the model predicts proportionally more false positives than false negatives. In other words, the prediction model makes more errors predicting negative edges as positive than vice versa.

For the second half of the plots, precision and recall are not meaningful metrics for evaluation due to the high number of true positives. We proceed by examining the false-positive rate (FPR), AUC ROC and the MCC metric, which can be found in Figure 7.4c, 7.4d and 7.4e. In general, the FPR is very high, which implies that an increased number of true negatives is wrongly predicted as positive. The negative edges are identified as false positives with a probability of approximately 90%, which is a very poor prediction. Moreover, the AUC ROC remains low at approximately 0.2 and 0.4, telling us that the model is unable to separate the true positives and true negatives. MCC reports a negative correlation with values between -0.5 and -0.2. This indicates that the model has a tendency to make inverse predictions, which is expected due to the high FPR and low TPR.

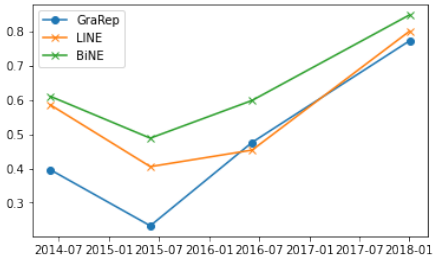
7.2.2 State of knowledge at V2

Taking the version at 2014.01.06 as the starting point of the prediction model, we end up with a disproportionally high number of positive edges. For this reason, we omit the evaluation of precision and recall and focus on the FPR, AUC ROC, and MCC metrics similar to above.

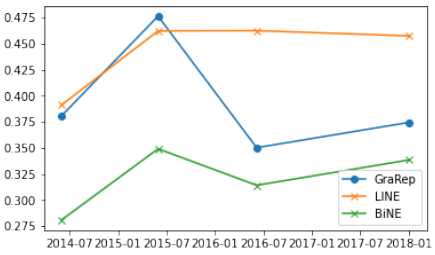
In Figure 7.5a, we notice that the FPR at t_1, t_2 and t_3 stays relatively low considering the very small number of negative edges (1-2% of total edges). However, at t_4, the FPR rises to 0.8, which implies that approximately 80% of the actual true negatives are predicted as positive. The negative edges at this point of time constitute around 10% of the total edges. This is significantly more than at the previous t_i, and from the high FPR, we can only deduce that the prediction model is unable to identify the true negatives correctly. The AUC ROC remains below 0.5, therefore we conclude that the model fails to predict the true positives and true negatives correctly. In Figure 7.5c, MCC is located between -0.2 and 0, which implies a weak correlation of inverse predictions. However, the closer it approaches 0, the more the predictions behave similarly to flipping a coin.

7.2.3 Discussion

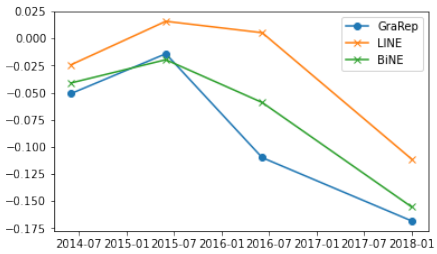
Comparing the performance of the two prediction models, we can confidently state that the version at 2014.01.06 performs better than at 2009.01.13. Taking the context of the domain into account, it is certainly beneficial to aim for a low FPR for the prediction



(a) FPR



(b) AUC ROC



(c) MCC

Figure 7.5: Performance metrics at 2014.01.06

model. Predicting an incorrect link between a drug and disease could be very detrimental compared with predicting a false negative that could have minor to no damage. The approximate average of the FPR at $V1$ is 0.86, whereas in $V2$ it decreases to 0.5. This means that the prediction of negative edges improves with $V2$, although they remain a minority. The AUC ROC value is also higher in V2 than in V1 moving up from approximately 0.3 to 0.4 and higher. However, everything below 0.5 is usually considered worse as this indicates an opposite behavior of what is expected. The same can be said about MCC, where we still observe a weak negative correlation that implies inverse predictions. Nevertheless, V1 reports a higher negative correlation than V2, and although the MCC value is now closer to 0, meaning that the prediction probability is similar to flipping a coin, it is better than having a higher possibility for inverse predictions. From this, we can infer that the prediction of future links improves with the evolution of the ontology. We observed further that the predictions of the LINE embeddings report better values and remain more stable compared to those of the GraRep and BiNE embeddings. In all the presented metrics, LINE outperforms the other two embedding methods, except for the FPR, where GraRep reports lower values than LINE. Nonetheless, the values of GraRep show more fluctuations, which is a sign of instability.

In this experiment, we are dealing with validated positive and negative samples; consequently, it differs from the link prediction task we presented in Section 7.1. This prediction task incorporates future links from a real-world setting. The predicted links are very likely to be imbalanced and may contain unforeseen relations between drugs and diseases. We have previously stated that similar drugs may treat similar diseases. Such an assumption might work in a small context in the real world, but we should

not restrict ourselves with this statement. The above arguments inevitably affect the prediction task's performance, which explains the overall low performance compared to Section 7.1.

8

Limitations

The published versions of the NDF-RT data set start from 2008 until the beginning of 2018, which in total adds up to 92 versions. However, with the introduction of a new generation named "NDF-RT2" in 2009, that includes a fundamental revision of the core concepts, we decided to omit the versions from 2008. The VHA restored several concepts and removed over 100'000 concepts with the new generation. Therefore, comparing the two different generations would not be beneficial due to the profound difference in content.

In terms of embedding methods, all of the used methods only require an edge list of known nodes as an input. The edge list and the meaning of a link between nodes can be defined by the user. This provides some freedom to decide on what we want to focus. However, it is also a restriction that brings along several limitations, two most crucial points being:

1. An embedding is only created for nodes that have at least one edge.
2. An embedding contains only neighborhood-related (one-dimensional) information.

To elaborate, (1) leads to the fact that one can conduct down-stream tasks such as link prediction only on known nodes. This means that e.g., a possible link can only be predicted between known nodes. Nodes without any edges are ignored in the prediction model as they have no embedding representation. This is quite a severe limitation and certainly not applicable in the real world. With (2), any additional node-related properties are not included in the embedding, which therefore restricts their usage for further tasks. The embeddings simply consist of neighborhood-related information. How the neighborhood is defined, depends on the edge context, which is given as input to the embedding method. In our case, the edges between the nodes stand for the *may treat* relation but it could for instance also represent *subclass of* or *parent of* relations.

In the neighborhood similarity task, we did not receive any results in lower percentiles ($r < 1$) for GraRep, which prevents us from making an absolute comparison of the embedding methods. Further, the distance metrics euclidean and cosine did not show a considerable difference in the results. We were thus unable to point out how the neighborhood similarity algorithm behaves with a different distance metric.

Regarding the link prediction task, the lack of validated negative samples restricted us from making more precise evaluations and statements regarding the performance.

With the current state, we can only assume that the logical restriction method filters out edges that are more likely to occur. There is no additional evaluation about the negative samples available.

To conclude, the neighborhood similarity and the link prediction task we conducted to determine an embedding's robustness should be regarded as a relative evaluation. Not only because of the prerequisites such a task involves but also because we are dealing with an evolving network.

9

Future Work

MED-RT is the continuation of NDF-RT; therefore, it is only a logical step to further analyze this data set and conduct experiments. As mentioned in Chapter 3 it would be beneficial to convert the XML structure of MED-RT into an OWL representation similar to what we did for the NDF-RT data set. Having the ontology in an OWL format makes it easier to analyze and observe the changes that are otherwise rather difficult to extract. In the same chapter we presented the categories of the drugs, respectively the diseases. With this additional information, we could examine how well the embeddings are clustered into the categories. Any (un)supervised clustering algorithm could be used to implement this further task.

In terms of embeddings, there are several ways to proceed. Intuitively, one could extend our work by using other embedding methods such as e.g., struc2vec [27] that focuses on the structural equivalence of nodes. There also exist several embedding methods for heterogeneous graphs [4, 6, 9], which could be beneficial to compare with BiNE. Another option would be to apply embedding methods that include node properties or hierarchical information such as owl2vec [5] which requires an OWL file as input. Utilizing any of these methods would certainly provide us with another perspective of the DDA network since the embedding methods we used heavily rely on the neighborhood. Another direction would be to focus on the bipartite nature of the DDA network primarily. For instance, adjusting the tasks to make them better applicable for bipartite graphs or even running separate tasks for the two node groups. Besides, the BiNE embeddings remain an unexplained subject, where we have seen several inconsistencies in the result of the neighborhood similarity task. On the one hand, we noticed a different pattern between the jaccard indexes and the overlap coefficients. On the other we received contrasting results for the two noise levels in the noise experiment. It would therefore be helpful to determine the cause of these inconsistencies to get a better understanding.

Furthermore, we defined robustness as receiving a consistent result of the tasks across the evolution. Consistency is determined by a non-significant mean difference between the results of two DDA versions. However, this definition of consistency should not be seen as complete. One could extend it by taking other metrics such as the median, the kurtosis, or the skewness of a distribution and compare it to another version's metric. To verify the significance of these metrics, we could perform a permutation test over the two distributions. Another perspective for robustness mentioned in our work is to analyze how the embeddings perform when adjusting a task's settings. In both tasks

we saw that certain embeddings react strongly to changes in the task, whereas others react little. Therefore, one could run the same tasks on another biomedical network with the respective embedding methods to verify if our observations can be confirmed. In general, it would be interesting to determine if we can reach the same conclusions with a different biomedical network.

10

Conclusion

Ontologies are essential in today's world, where data has become an indispensable commodity. Transforming their content into embeddings makes it possible to analyze and run prediction models. This helps us to simplify or solve tasks in the real-world that would otherwise be very expensive when done manually. For instance, link prediction saves a lot of time when having to find associations between two entities. Although it is not feasible to fully replace a human evaluator, it would still reduce preliminary work so that in the best-case scenario, a human evaluator has to go through the prediction results only to confirm their validity. This is just one of the countless application methods embeddings can be used for.

In this thesis, we chose several versions of an evolving DDA network and transformed them into embeddings. After that, we applied three different embedding methods and ran several tasks with them. Comparing the results of these tasks across the evolution helped us to determine the robustness of an embedding. Two major tasks performed were the local neighborhood similarity and link prediction. The first research question concerns the neighborhood similarity task, where we wanted to investigate how consistent the result of the embeddings are in terms of evolution. To answer this question, we have computed the similarities within the versions with different distance (percentile) values, which imply a threshold for the neighborhood. This was followed by a comparison with the other versions' results. The neighborhood similarity showed that the embeddings report an overall low similarity of approximately 0.6 when comparing one version with different seeds. While matching the similarity metrics with those of the other versions, we found that only the neighboring versions show no significant difference in the mean. The remaining versions report significance due to the inherent nature of the embeddings or larger changes in the ontology. We further noticed a turning point (2014.06.02) for certain percentile values that coincides with a larger change in the ontology dividing the versions into two groups. In general, the different percentile values on any embedding method showed an inconsistent behavior where at certain percentiles, several versions proved no significance, and at others, a majority of significantly different means emerged. Among the three embedding methods, LINE proved to be the most consistent where we were able to observe a more or less consistent pattern across the percentile values. To answer ***RQ1***, the results showed that the embedding methods could not guarantee a consistent result across the evolution with the different percentile values. Only with

respect to near-term evolution, the embedding methods were able to output a stable result.

The second research question involves the link prediction task. We wanted to find how the evolution influences the prediction's performance. To answer this question, we performed the task on each version of the evolution and then compared the AUC ROC value with the other versions' AUC ROC. The baseline was approximately 0.85 and with the logical restriction method for the negative sample selection we were able to achieve an AUC ROC of around 0.95. For this, we changed the settings of the link prediction algorithm which as a result exerted an influence on the results' consistency across the evolution. GraRep and BiNE showed a more consistent performance metric across the evolution with the *0-step* and *3-step* method; however, this changes drastically when considering the *7-step* method, where especially for GraRep, most of the versions got rejected. For LINE, we identified the same turning point as seen in the neighborhood similarity, thus the consistency across the evolution is only ensured for neighboring versions. Nonetheless, we still observed a consistent pattern over the different step sizes from which we can conclude that LINE is more stable against modifications in both tasks. Again, it was proven that a larger change in the ontology leads to a shift in the mean where two groups with different mean can be determined. With the above-mentioned turning point, we could argue that LINE is more sensitive to changes in the ontology than GraRep or BiNE. However, larger changes in the ontology inevitably have an impact on the subsequent tasks and thus are not avoidable. To answer ***RQ2***, only the neighboring versions ensure a consistent performance across the evolution for any embedding method. However, when considering the performances across the different step sizes, LINE outperforms the other two methods.

Our work shows that the comparison of embeddings is non-trivial and highly dependent on the embedding method and the parameter settings of a task. It also depends on how disparate the compared embeddings are, as we found that neighboring versions usually report no significant difference, unlike farther apart versions. We redefined our understanding of robustness as it not only refers to how stable the embeddings perform across the evolution but also how they react to certain changes in the task. Nonetheless, these conclusions remain in the context of the DDA network and should be further applied to other ontologies for confirmation.

References

[1] Bodenreider, O. The Unified Medical Language System (UMLS): integrating biomedical terminology. *Nucleic Acids Research 32*, suppl_1 (01 2004), D267–D270.

[2] Boggust, A., Carter, B., and Satyanarayan, A. Embedding comparator: Visualizing differences in global structure and local neighborhoods via small multiples, 2019.

[3] Cao, S., Lu, W., and Xu, Q. Grarep. pp. 891–900.

[4] Chang, S., Han, W., Tang, J., Qi, G.-J., Aggarwal, C. C., and Huang, T. S. Heterogeneous network embedding via deep architectures. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (New York, NY, USA, 2015), KDD '15, Association for Computing Machinery, pp. 119-128.

[5] Chen, J., Hu, P., Jimènez-Ruiz, E., Holter, O., Antonyrajah, D., and Horrocks, I. Owl2vec*: Embedding of owl ontologies, 09 2020.

[6] Chen, T., and Sun, Y. Task-guided and path-augmented heterogeneous network embedding for author identification. WSDM '17, Association for Computing Machinery, pp. 295-304.

[7] Dai, W., Liu, X., Gao, Y., Chen, L., Song, J., Chen, D., Gao, K., Jiang, Y., Yang, Y., and Chen, J. Matrix factorization-based prediction of novel drug indications by integrating genomic space. *Computational and Mathematical Methods in Medicine 2015* (06 2015), 1–9.

[8] Ding, D., Zhang, M., Pan, X., Yang, M., and He, X. Improving the robustness of wasserstein embedding by adversarial pac-bayesian learning. *Proceedings of the AAAI Conference on Artificial Intelligence 34* (04 2020), 3791–3800.

[9] Dong, Y., Chawla, N. V., and Swami, A. Metapath2vec: Scalable representation learning for heterogeneous networks. KDD '17, Association for Computing Machinery, pp. 135-144.

[10] Flouris, G., Manakanatas, D., Kondylakis, H., Plexousakis, D., and Antoniou, G. Ontology change: Classification and survey. *Knowl. Eng Rev. 23*, 2 (2008), 117–152.

[11] Gao, M., Chen, L., He, X., and Zhou, A. Bine: Bipartite network embedding. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval* (New York, NY, USA, 2018), SIGIR '18, Association for Computing Machinery, pp. 715-724.

[12] Gottlieb, A., Stein, G. Y., Ruppin, E., and Sharan, R. Predict: a method for inferring novel drug indications with application to personalized medicine. *Molecular systems biology 7* (June 2011), 496.

[13] Goyal, P., Huang, D., Goswami, A., Chhetri, S. R., Canedo, A., and Ferrara, E. Benchmarks for graph embedding evaluation, 2019.

[14] Gross, A., Hartung, M., Prüfer, K., Kelso, J., and Rahm, E. Impact of ontology evolution on functional analyses. *Bioinformatics 28*, 20 (2012), 2671–2677.

[15] Hamilton, W. L., Leskovec, J., and Jurafsky, D. Cultural shift or linguistic drift? comparing two computational measures of semantic change. *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing 2016* (November 2016), 2116–2121.

[16] Hasan, M., Chaoji, V., Salem, S., and Zaki, M. Link prediction using supervised learning.

[17] Hinton, G., and Roweis, S. Stochastic neighbor embedding. In *Advances in Neural Information Processing Systems 15* (2003), MIT Press, pp. 833–840.

[18] Kulmanov, M., Liu-Wei, W., Yan, Y., and Hoehndorf, R. EL embeddings: Geometric construction of models for the description logic EL ++. *CoRR abs/1902.10499* (2019).

[19] Liang, X., Zhang, P., Yan, L., Fu, Y., Peng, F., Qu, L., Shao, M., Chen, Y., and Chen, Z. LRSSL: predict and interpret drug–disease associations based on data integration using sparse subspace learning. *Bioinformatics 33*, 8 (01 2017), 1187–1196.

[20] Matthews, B. Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA) - Protein Structure 405*, 2 (1975), 442–451.

[21] McInnes, L., and Healy, J. Umap: Uniform manifold approximation and projection for dimension reduction.

[22] Motik, B., Patel-Schneider, P. F., and Grau, B. C. Owl 2 web ontology language: Profiles. recommendation, world wide web consortium (w3c).

[23] Orme, A. M., Yao, H., and Etzkorn, L. H. Indicating ontology data quality, stability, and completeness throughout ontology evolution. *J. Softw. Maint. Evol. Res. Pract. 19*, 1 (2007), 49–75.

[24] Pearson, K. F. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science 2*, 11 (1901), 559–572.

[25] Pernischova, R., Dell'Aglio, D., Horridge, M., Baumgartner, M., and Bernstein, A. Towards Predicting Impact of Changes in Evolving Knowledge Graphs. In *Posters & Demonstrations* (Aukland, NZ, Oct. 2019), Springer.

[26] Postell, W. D. Medicines for the union army. *Bulletin of the Medical Library Association 51*, 1 (January 1963), 145-146.

[27] Ribeiro, L. F., Saverese, P. H., and Figueiredo, D. R. Struc2vec: Learning node representations from structural identity. KDD '17, Association for Computing Machinery, pp. 385-394.

[28] Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., and Mei, Q. Line: Large-scale information network embedding. WWW '15, International World Wide Web Conferences Steering Committee, pp. 1067-1077.

[29] TensorFlow. Embeddings. *https://www.tensorflow.org/guide/embedding*, 2021. Accessed: 2021-01-31.

[30] Wang, Q., Mao, Z., Wang, B., and Guo, L. Knowledge Graph Embedding: A Survey of Approaches and Applications. *IEEE Trans Knowl Data Eng 29*, 12 (2017), 2724–2743.

[31] Wu, G., and Liu, J. Predicting drug-disease treatment associations based on topological similarity and singular value decomposition. In *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)* (2019), pp. 153–158.

[32] Yue, X., Wang, Z., Huang, J., Parthasarathy, S., Moosavinasab, S., Huang, Y., Lin, S. M., Zhang, W., Zhang, P., and Sun, H. Graph embedding on biomedical networks: methods, applications and evaluations. *Bioinformatics 36*, 4 (2020), 1241–1251.

[33] Zeng, K., Kilbourne, J., Powell, T., and Moore, R. Normalized names for clinical drugs: Rxnorm at 6 years. *JAMIA 18* (07 2011), 441–8.

[34] Zhang, W., Yue, X., Chen, Y., Lin, W., Li, B., Liu, F., and Li, X. Predicting drug-disease associations based on the known association bipartite network. In *2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)* (2017), pp. 503–509.

[35] Zhang, W., Yue, X., Lin, W., Wu, W., Liu, R., Huang, F., and Liu, F. Predicting drug-disease associations by using similarity constrained matrix factorization. *BMC Bioinformatics 19* (06 2018).

A

Appendix

A.1 Extract from OWL file

Figure A.1 shows an extract of the NDF-RT data set's OWL representation. In A.1a, a drug entity is depicted where we can see several information such as the name and the NUI as well as relations to the parent drug and to a disease. In A.1b, the disease entity is shown with several information such as the name, NUI and description.

A.2 Hyperparameters

We introduce the hyperparameters used for the embedding generation in Table A.1. The parameters for GraRep and LINE were taken from Yue et al. [32], where they conducted an in-depth tuning. Similarly, the parameters for BiNE were tuned in by Gao et al. [11] by measuring the link prediction performance. The default parameters were taken for those not mentioned in A.1.

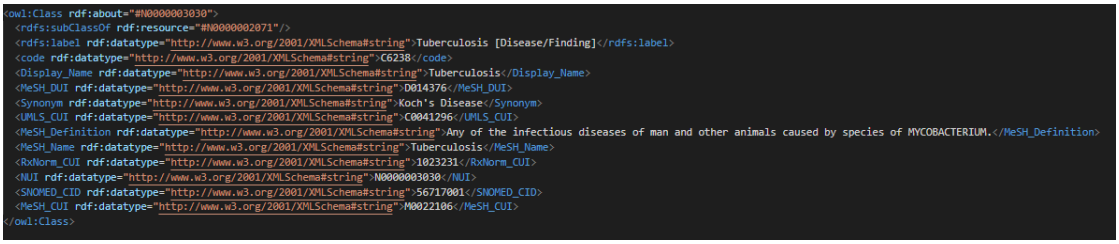
Method	Dimensions	Parameters
GraRep	100	kstep = 4, weight-decay = 5e-4, lr = 0.01
LINE	100	epochs = 10, lr = 0.01, negative-ratio = 5, order = 2
BiNE	100	$\alpha = 0.01$, $\beta = 0.01$, $\gamma = 10$

Table A.1: Hyperparameters for embedding methods

A.3 Dimension Reduction Techniques

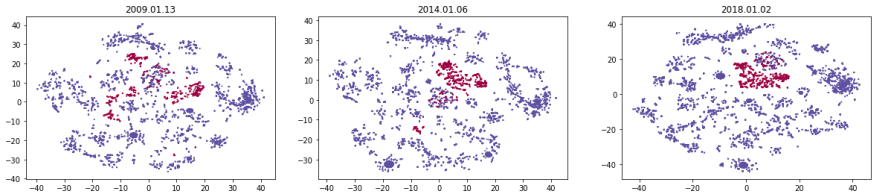
The two-dimensional projections of the embeddings with t-SNE and UMAP are presented in this section. Figure A.2 and A.3 depict the respective embeddings and we notice that for the GraRep embeddings the cluster tendency increases along the evolution or rather is preserved. In contrast, the drug points and disease points for LINE show no cluster tendency in t-SNE and only in UMAP we can observe that the drugs are centralized in the middle. BiNE demonstrates with t-SNE a similar pattern to PCA. In UMAP, there are no clusters visible as the points are scattered miscellaneously.

(a) Drug entity in OWL

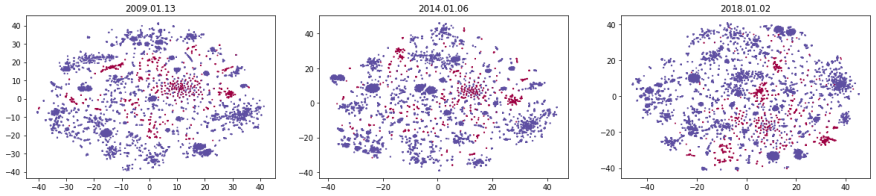


(b) Disease entity in OWL

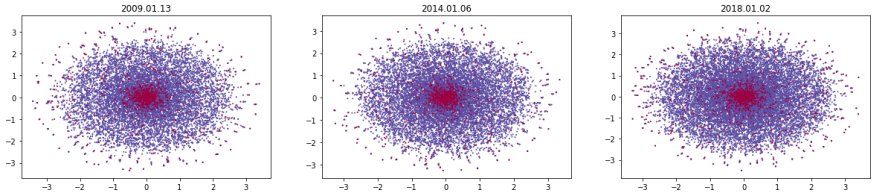
Figure A.1: OWL Representation



(a) GraRep

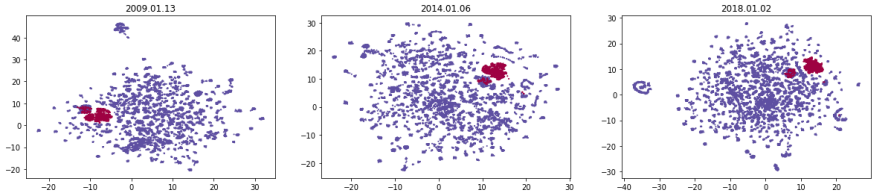


(b) LINE

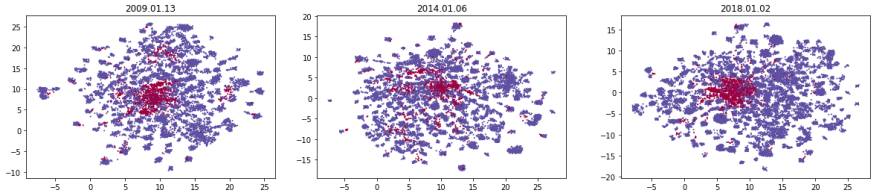


(c) BiNE

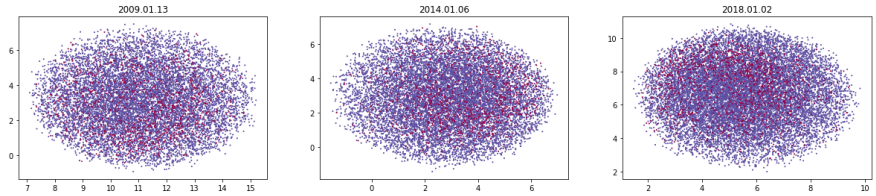
Figure A.2: t-SNE on the evolution (seed=5). Drugs are purple and Diseases are red.



(a) GraRep



(b) LINE



(c) BiNE

Figure A.3: UMAP on the evolution (seed=5). Drugs are purple and Diseases are red.

A.4 Neighborhood Similarity

As we have observed in several cases, the euclidean and cosine distance behave similarly and thus output similar results. We present here the results of the cosine distance, which were omitted in the actual report.

A.4.1 Evolution-oriented

Figure A.4 presents the neighborhood sizes with the cosine distance of the three embedding methods with different percentile values. Here as well, we notice that with a higher percentile value, the neighborhood sizes become more similar. We can also observe an increase in the neighborhood as the ontology evolves.

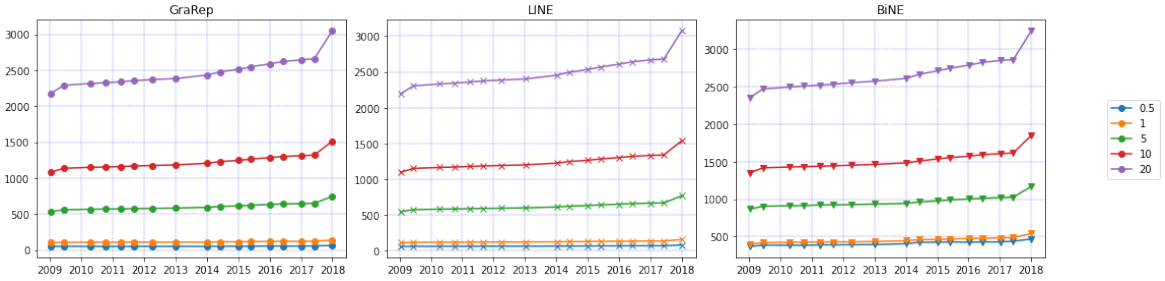
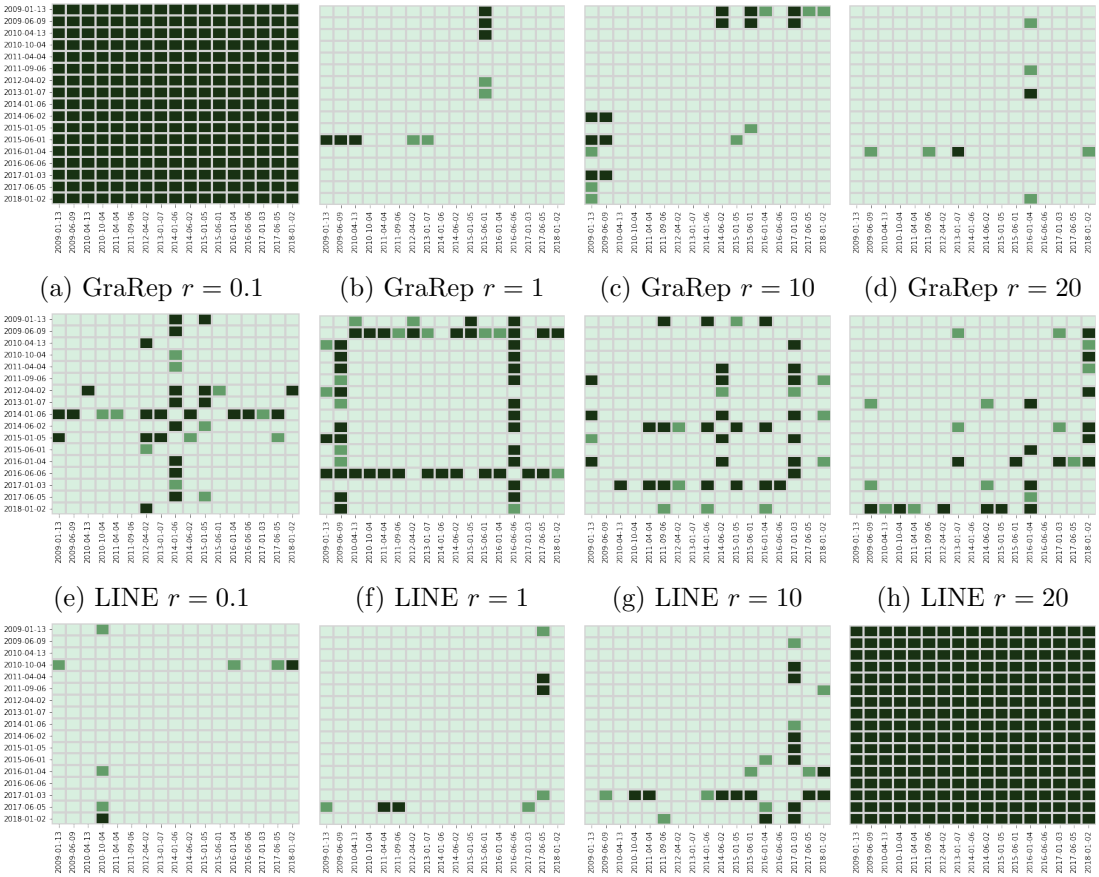


Figure A.4: Neighborhood sizes with cosine distance

In Figure A.5, the result of *Levene's t-test* with the euclidean distance is presented. This test was conducted to determine which statistical test for mean comparison suits the data set. Figure A.6 and A.7 demonstrate the result of *Levene's test* and *Welch's t-test* with the cosine distance. *Welch's t-test* was used as this test does not require equal variance between two samples. The pattern and consequently the results show almost no difference to the one presented in the report; thus, we can treat this as a duplicate.

A.4.2 Noise-oriented

Similar to above, we first present the results of *Levene's test* for the euclidean and the cosine distance in Table A.2 and A.3. The results of the two distance metrics show a significant different variance for a few cases. Nonetheless, the values for the euclidean and cosine distance are very similar. We proceed with using *Welch's t-test* in order to compare the means between two groups. The result for the cosine distance can be found in Table A.4. All comparisons report a significant difference in the mean of the ground-truth to the synthetic versions. This is once again similar to the results we received for the euclidean distance.

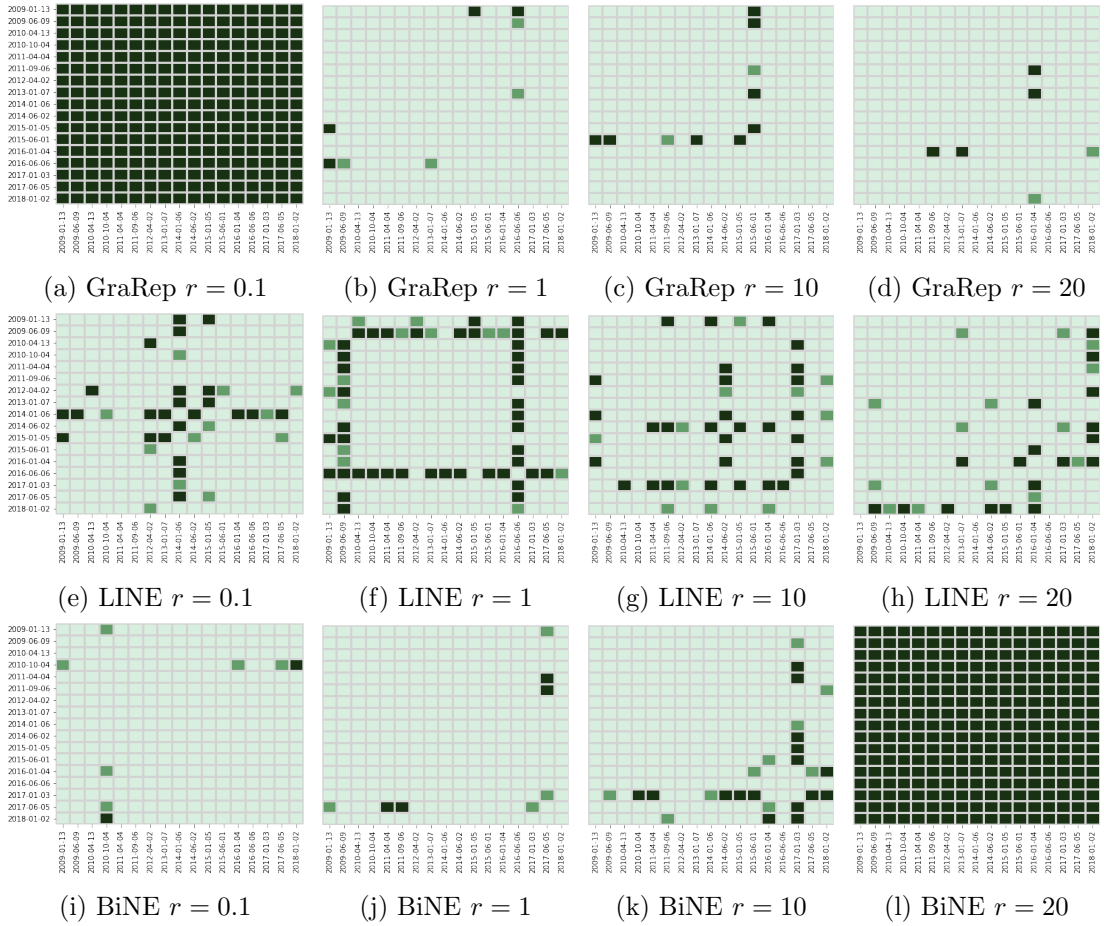


(a) GraRep $r = 0.1$ (b) GraRep $r = 1$ (c) GraRep $r = 10$ (d) GraRep $r = 20$

(e) LINE $r = 0.1$ (f) LINE $r = 1$ (g) LINE $r = 10$ (h) LINE $r = 20$

(i) BiNE $r = 0.1$ (j) BiNE $r = 1$ (k) BiNE $r = 10$ (l) BiNE $r = 20$

Figure A.5: Levene's test for LN similarity comparison with euclidean distance (dark green stands for rejected null hypothesis and light green for accepted null hypothesis)

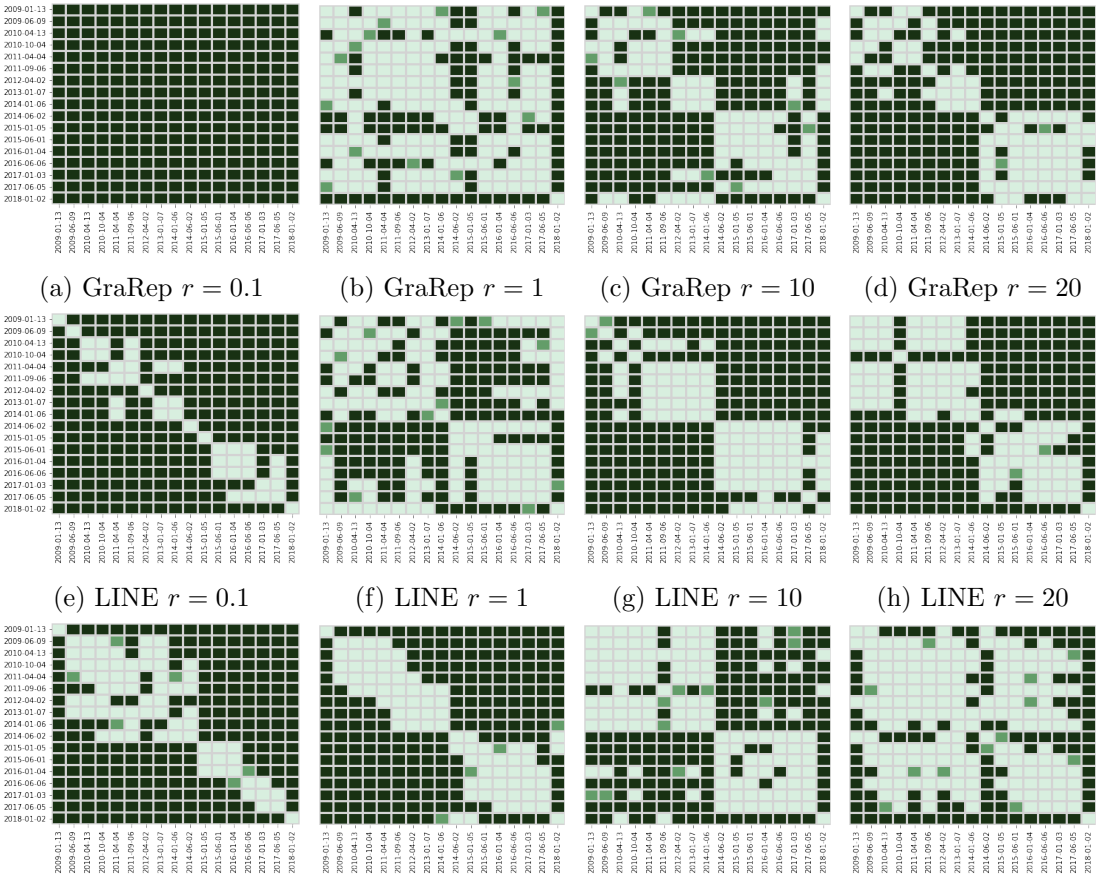


(a) GraRep $r = 0.1$ (b) GraRep $r = 1$ (c) GraRep $r = 10$ (d) GraRep $r = 20$

(e) LINE $r = 0.1$ (f) LINE $r = 1$ (g) LINE $r = 10$ (h) LINE $r = 20$

(i) BiNE $r = 0.1$ (j) BiNE $r = 1$ (k) BiNE $r = 10$ (l) BiNE $r = 20$

Figure A.6: Levene's test for LN similarity comparison with cosine distance (dark green stands for rejected null hypothesis and light green for accepted null hypothesis)



(a) GraRep $r = 0.1$ (b) GraRep $r = 1$ (c) GraRep $r = 10$ (d) GraRep $r = 20$

(e) LINE $r = 0.1$ (f) LINE $r = 1$ (g) LINE $r = 10$ (h) LINE $r = 20$

(i) BiNE $r = 0.1$ (j) BiNE $r = 1$ (k) BiNE $r = 10$ (l) BiNE $r = 20$

Figure A.7: Welch's t-test for LN similarity comparison with cosine distance (dark green stands for rejected null hypothesis and light green for accepted null hypothesis)

	GraRep		**LINE**		**BiNE**	
	statistic	p-value	statistic	p-value	statistic	p-value
$cos_{<0.1\%}$						
GT − Noise L1	-	-	1.5698	<.0001	1.8047	0.1958
GT − Noise L2	-	-	0.2144	<.0001	0.5480	0.4687
$cos_{<1\%}$						
GT − Noise L1	1.1853	0.2800	1.6487	0.2034	0.1767	0.6792
GT − Noise L2	1.4791	0.2280	5.9352	0.0174	<.0001	0.9943
$cos_{<10\%}$						
GT − Noise L1	1.8960	0.1729	0.0149	0.9032	0.0035	0.9535
GT − Noise L2	4.5685	0.0361	5.0789	<.0001	1.1974	0.2882
$cos_{<20\%}$						
GT − Noise L1	0.0038	0.9513	<.0001	0.9963	3.1164	0.0945
GT − Noise L2	0.8330	0.3646	2.0306	0.1586	0.1785	0.6777

Table A.2: Levene's test for the noise experiment with the euclidean distance.

	GraRep		**LINE**		**BiNE**	
	statistic	p-value	statistic	p-value	statistic	p-value
$cos_{<0.1\%}$						
GT − Noise L1	-	-	1.5977	0.2104	2.0282	0.1715
GT − Noise L2	-	-	0.0253	0.8742	0.3897	0.5403
$cos_{<1\%}$						
GT − Noise L1	0.5149	0.4754	1.6056	0.2093	0.1900	0.6682
GT − Noise L2	0.0492	0.8251	5.9138	0.0176	0.0001	0.9919
$cos_{<10\%}$						
GT − Noise L1	2.3803	0.1274	0.0153	0.9020	0.0037	0.9520
GT − Noise L2	5.3123	0.0241	5.0714	0.0275	1.1831	0.2911
$cos_{<20\%}$						
GT − Noise L1	0.0176	0.8948	<.0001	0.9972	3.1623	0.0923
GT − Noise L2	1.1079	0.2962	2.0326	0.1584	0.1825	0.6743

Table A.3: Levene's test for the noise experiment with the cosine distance.

A.5 Link Prediction Results

We present the results of *Levene's test* for the evolution-oriented and the noise-oriented approach that were omitted in the report.

A.5.1 Evolution-oriented

In Figure A.8, the results of *Levene's test* are depicted. This test was conducted to determine which statistical test for mean comparison suits the data set. GraRep shows only in few cases a significant difference in variance. In contrast, LINE rejects the null

	GraRep		LINE		BiNE	
	statistic	p-value	statistic	p-value	statistic	p-value
$cos_{<0.1\%}$						
GT − Noise L1	-	-	7.7662	<.0001	6.5333	<.0001
GT − Noise L2	-	-	15.7306	<.0001	13.2793	<.0001
$cos_{<1\%}$						
GT − Noise L1	32.6752	<.0001	31.6711	<.0001	5.7264	<.0001
GT − Noise L2	44.0721	<.0001	77.9432	<.0001	3.9293	0.0009
$cos_{<10\%}$						
GT − Noise L1	55.1386	<.0001	10.7482	<.0001	10.7555	<.0001
GT − Noise L2	81.5746	<.0001	16.6312	<.0001	5.5428	<.0001
$cos_{<20\%}$						
GT − Noise L1	53.4176	<.0001	5.8053	<.0001	33.4305	<.0001
GT − Noise L2	89.2832	<.0001	11.4750	<.0001	16.0497	<.0001

Table A.4: Welch's t-test for the noise experiment with the cosine distance.

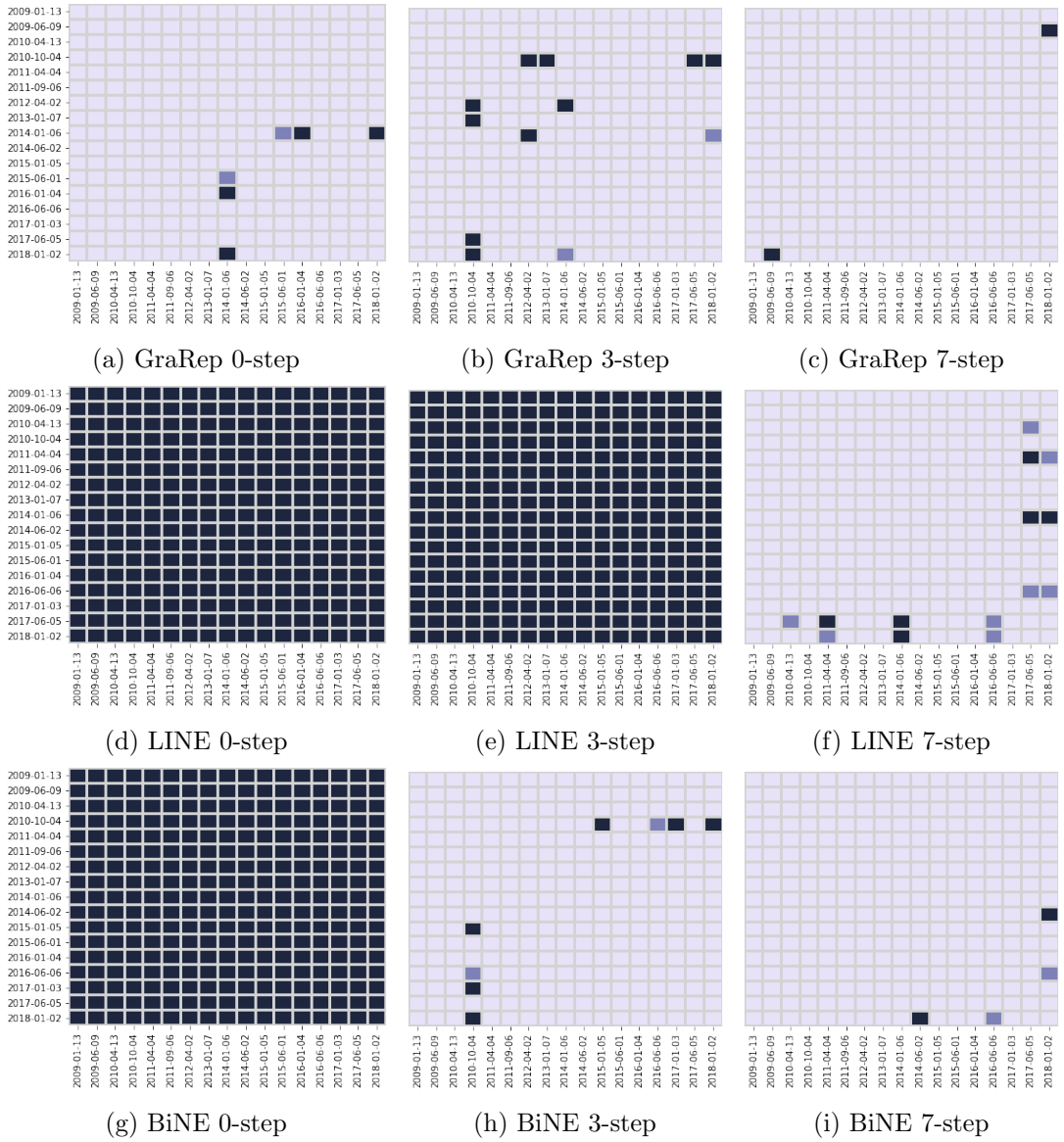
hypothesis for all versions with the *0-step* and *3-step* method. Only in the *7-step* method, a few versions are rejected. BiNE reports only few cases of rejection in *3-step* and *7-step* method. However, in the *0-step* method, all versions reject the null hypothesis. Due to the inconsistent results, we have decided to use the *Welch's t-test* that does not require equal variance between two samples.

A.5.2 Noise-oriented

Table A.5 presents the result of the *Levene's test* for the noise experiment. For all embedding methods and step sizes, the test reported an equal variance. This means that despite the noise addition and the different step size the variance of the results remains the same.

	GraRep		LINE		BiNE	
	statistic	p-value	statistic	p-value	statistic	p-value
0-step						
GT − Noise L1	0.1330	0.7196	0.0541	0.8187	0.3855	0.5520
GT − Noise L2	2.4551	0.1346	1.2115	0.2855	0.0782	0.7869
3-step						
GT − Noise L1	0.1365	0.7161	0.1366	0.7160	0.4823	0.5071
GT − Noise L2	3.2331	0.0890	0.0035	0.9538	0.0099	0.9229
7-step						
GT − Noise L1	1.0322	0.3231	2.4107	0.1379	2.3399	0.1646
GT − Noise L2	0.1075	0.7467	2.0383	0.1705	2.6038	0.1453

Table A.5: Levene's test (AUC ROC) for the noise experiment



(a) GraRep 0-step (b) GraRep 3-step (c) GraRep 7-step

(d) LINE 0-step (e) LINE 3-step (f) LINE 7-step

(g) BiNE 0-step (h) BiNE 3-step (i) BiNE 7-step

Figure A.8: Levene's test with AUC ROC.

List of Figures

List of Tables