UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

EMBEDDING DATASETS FOR EXPLAINABLE MALWARE DETECTION

Master thesis

2022 Bc. Daniel Trizna

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

EMBEDDING DATASETS FOR EXPLAINABLE MALWARE DETECTION

Master thesis

Študijný program:	Aplikovaná informatika
Študijný odbor:	2511 Aplikovaná informatika
Školiace pracovisko:	Katedra aplikovanej informatiky
Školiteľ:	doc. RNDr. Martin Homola, PhD.

Bratislava, 2022

Bc. Daniel Trizna

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Bc. Daniel Trizna
Študijný program: aplikovaná informatika (Jednoodborové štúdium, magisterský II. st., denná forma)
Študijný odbor: informatika
Typ záverečnej práce: diplomová
Jazyk záverečnej práce: anglický
Sekundárny jazyk: slovenský

Názov: Embedding datasets for explainable malware detection
Znalostná vektorizácia datasetov pre vysvetliteľnú detekciu malvéru

Anotácia: Znalostná vektorizácia je spôsob prekladu symbolických dát do vektorových priestorov. Na takto vektorizované možno aplikovať strojové učenie, či iné subsymbolové metódy, pričom preklad zaručuje zachovanie podstatných sémantických relácií a takto obohatené dáta môžu byť opäť symbolicky interpretované. Táto práca skúma možné aplikácie metód znalostnej vektorizácie v oblasti informačnej bezpečnosti.

Cieľ: 1) Prehľad literatúry z oblasti vhodných metód znalostnej vektorizácie
2) Výber 1-2 kandidátskych metód
3) Výber vhodného datasetu v oblasti malvéru/informačnej bezpčnosti
4) Aplikácia na problém vysvetliteľnej detekcie malvéru/bezpečnostných prienikov

Literatúra: Badreddine, S., Garcez, A.D.A., Serafini, L. and Spranger, M., 2021. Logic tensor networks. Artificial Intelligence, p.103649.
Bordes et al., 2013. Translating embeddings for modeling multi-relational data. Advances in neural information processing systems, 26.
Husák, M., Komárková, J., Bou-Harb, E. and Čeleda, P., 2018. Survey of attack projection, prediction, and forecasting in cyber security. IEEE Communications Surveys & Tutorials, 21(1), pp.640-660.
Özçep, et al., 2021. Cone semantics for logics with negation. In Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence (pp. 1820-1826).

Vedúci: doc. RNDr. Martin Homola, PhD.
Katedra: FMFI.KAI - Katedra aplikovanej informatiky
Vedúci katedry: prof. Ing. Igor Farkaš, Dr.

Dátum zadania: 14.12.2021

Dátum schválenia: 16.12.2021

prof. RNDr. Roman Ďurikovič, PhD.
garant študijného programu

....
študent

....
vedúci práce

I hereby declare that this master thesis is written by me.

. . . .

Bratislava, 2022

Bc. Daniel Trizna

Acknowledgement

Abstract

Keywords:

Abstrakt

Kľúčové slová:

Contents

Chapter 1

Introduction

In this chapter we will introduce reader to basic concepts and approaches that are known and usefull for goals of our masters thesis.

In first part of this chapter we will go trough what explainable malware detection is and how it is diferent from classical malware detectoin. In second part of this chapter we will introduce reader to basic concepts of knowledge base embedding as well as some basic approaches.

1.1 Introduction to Malware Detections

In this section we will introduce reader to basic concepts of malware detection as well as difference between explainable malware detection and basic malware detection.

1.1.1 Basic Malware Detection

By definition malware detection is **TODO dat definicio malware detectionu plus dat zdroj**

1.1.2 Explainable Malware Detection

The main difference between basic malware detection and explainable malware detection is in explainability. We try to not only provide answer whether the file is malware or not but also answer for question why is the file considered malware and also why not. Other main requirement for exlpainable malware detection is that output should be readable for humans not only for computers.

1.2 Introduction to Knowledge Base Embedding

In this section we will explain our reader what is knowledge base. We will also look at some basic goals that knowledge base embedding is trying to acheive as well as some basic approaches.

1.2.1 What is Knowledge Base

There are multiple ways to store our knowledge base. In this subsection we will look only on two basic formats of knowledge bases that are relevant for our work. Firstly we will take a look at knowledge graphs, that are less expressionable and then we will look at onthologies.

Knowledge Graphs

Knowledge graphs are simple but effective way to store our knowledge bases. In knowledge graphs we express each class as one node and edges between these nodes expresses relations between the classes. The main advantage of this approach of storing knowledge bases is that we can run all graphs algorithms on the knowledge base which can help with querying and also comparing two knowledge bases based on some bipartite criteria. Another advantage is that it is simple and straight forward format that is easy to understand and visualize. The main disadvantage of this approach is of course lack of expressivity. It is fairly hard to express complex relations in this format and relations like subsumption and union are nearly impossible to express.

Ontology

Storing knowledge bases in ontologies provides us with much more expressivity. There are multiple different ontology languages that were created to improve expressivity as time passes. The main languages and their differences are shown in table **TODO dat tablku porovnania ontologii**. As we can see the most expressive one is OWL2, which is also beeing used in ontology we are experimenting with.

Chapter 2

Motivation

Chapter 3

Issues overview

In this chapter we wil take a look at issues that we will encounter while working on solution for the explainable malware detection using knowledge base embedding.

We can split these issues into two cathegories which are issues concerning the knowledge base embedding and knowledge base itself and second cathegory is explainable malware detection itself.

1. Knowledge base embedding

2. Explainable malware detection

3.1 Knowledge Base Embedding

Knowledge base embedding as explained in previous chapter is transforming knowledge base in form of ontology or knowlege graph into some vector space. The main problems that knowledge base embedding has to solve are:

- geometrical representation

- information loss

3.1.1 Geometrical representation

The main question for knowledge base embedding is whether to embed classes into some geometrical representation or just simply keep them unorganized. As study shown **TODO dat tu referenciu na clanok o geometrical knowledge base embeddingu** much better results are aquired when embedding classes and their relationships into some geometrical objets.
To resolve this issue we will have to explore approaches that tries to embed datasets into geometrical objects as well as those that just tries to embed them withou any specific representation.

3.1.2 Information Loss

Whenever we transfer datasets into vector space there is always possibility that the new representation will not fully reflect all the information that were available in previos representation. The main aim of every knowledge base embedding is to keep as much information as possible when transfering the knowledge base into new vector space. For embeddings that embed classes in geometric objects this task is much easier. For example to express that two classes are disjoint the embedding simply create two geometric objects in such way that they have no intersection. In common fashon it is possible to embed also intersection of two classes as well as conjunction.

3.2 Explainable Malware Detection

In this section we will take a look at what possible issues we will be facing when trying to extract information whether the file is malware or not and also why is it categorized as a malware from our knowledge base embedding. The main problems that we will have to solve are:

- classification
- explanation

3.2.1 Classification

Once we have functioning knowledge base embedding we have to measure its accuracy. Accuracy of our trained embedding should be measured based on how well we can classify provided file to two categories whether the file is malware or not.

3.2.2 Explanation

Other issue that needs to be resolved is to provide human readable and understanable explanation why the file was classified as malware or as nonmalware. So for this problem we will have to solve two following issues:

- Extraction of explenation why the file was classified as malware or nonmalware.
- Human readable explenation.

Chapter 4

Previous Solutions

In knowledge base embedding there are multiple approaches, some better than others. In this chapter we will take a look at few approaches to knowledge base embedding. We will look at their positive and negative properties as well as their usability in our domain - explainable malware detection.

4.1 TransE embedding

First embedding that we will look at is TransE embedding. This is one of first embeddings that have been used. TransE falls into category of embeddings that does not embed classes nor relations into any specific geometrical object.

4.1.1 What knowledge base does TransE support

TransE is one of the simplier embeddings and as a result it does not support ontologies nor any class assumptions such as intersection subsumption or disjunction. As a result TransE requires dataset in form of knowledge graphs.

4.1.2 How does TransE work

TransE as mentioned is fairly simple. TransE training algorithm tries to minimize margin based error which is computed by sum provided on image 4.1.

$$\sum_{(h,\ell,t)\in S}\ \sum_{(h',\ell,t')\in S'_{(h,\ell,t)}} \left[\gamma + d(\boldsymbol{h}+\boldsymbol{\ell},\boldsymbol{t}) - d(\boldsymbol{h'}+\boldsymbol{\ell},\boldsymbol{t'})\right]_+$$

Figure 4.1: Margin based error

In the previously mentioned sum (fig. 4.1) letters *h* and *t* stands for two classes that are in relation *l* and letters *h'* and *t'* stands for two classes that are chosen in such way that they are not in relation *l*. Simply put the training algorithm is trying to achieve that the distance between classes that are in relation *l* is smaller than distance between random classes that are not in that relation. So for each triples (*h*, *l*, *t*) and any triples (*h'*, *l*, *t'*) that are chosen in a way that *h'* is not in relation *l* with *t'* should hold

$$d(h + l, t) < d(h' + l, t')$$

4.1.3 Conclusion

In this subsection we will go trough some positives and negatives that comes with usage of TransE embedding in our domain.

Positives

Main positive of TransE is that it is simple and fairly fast to train. TransE is also very good baseline for comparison of recall ability of knowledge base embedings.

Negatives

Because of the fat that TransE does not support any class assumptions it causes dataset to lose its expresivity. Another big drawback of TransE is that it can not embed data point that was not available in training process. This makes validation and testing more complicated but not impossible.

Usability

TransE approach will be in our domain usable only as a baseline for comparing other knowledge base embeddings. Main issue is that since TransE does not embed classes into any specific geometrical objects we can not extract any reasonable explanation to find out why was some file labeled as malware or non-malware.

4.2 Sphere embedding

Another embedding that we will explore is embedding of *EL++* logic. This approach is much more sophisticated than previously mentioned TransE embedding. In this approach as name suggests we will be embedding classes and relation into convex n-balls (spheres).

4.2.1 What knowledge base does Sphere embedding support

As mentioned before this embedding works with ontology written in *EL++* language. This language and its expresivity is worse than OWL2 but we can at least reflect some class assertions as subsumption intersection and disjunction.

4.2.2 How does Sphere embedding work

Sphere embedding algorithm represents its embedding with tuple of two functions:

$$f : C \cup R \rightarrow \mathbb{R}^n$$
$$r : C \rightarrow \mathbb{R}$$

First function f for every class gives vector which represents center of sphere into which is the class going to be embedded and for each relation its embedding vector.

Second function r for each class assigns one number representing the radius of the sphere into which the specific class is going to be embedded.

During training algorithm both of these functions are trained to most successfully fit provided dataset.

On image 4.2 we can see how does the Sphere embedding works on simple family onthology in two dimensional space.

4.2.3 Conclusion

In this subsection we will go trough some positives and negatives that comes with usage of Sphere embedding in our domain.

Positives

Main positive of Sphere embedding is that it embeds classes into separate geometric objects so that we can then also extract explanation why is file classified as malware. Sphere embedding supports more complex class as-

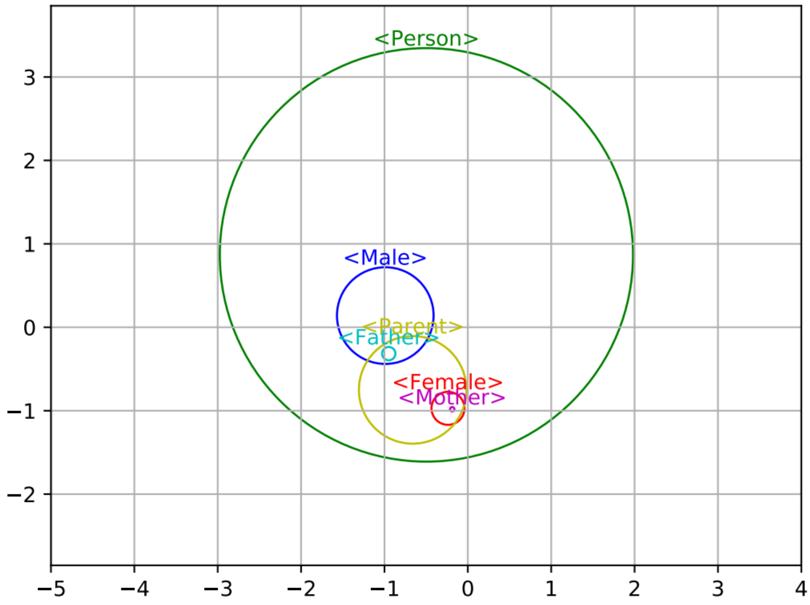

Figure 4.2: Sphere embedding on simple family ontology [3]

sumptions which helps us to keep as many information as possible present in final embedding.

Negatives

The main negative of this approach is that we the source codes provided by authors does not work out of box and are not compatible with any other dataset than the one used in the reaserch.

Usability

Sphere embedding should yield good results and should be useful for explainability of the malware detection.

4.3 Box embedding

Next embedding we will look into is BoxEL embedding. This is most recent embedding approach. This is also as previous embedding embedding which aims to embed knowledge base into geometrical objects. As name suggests this embedding is also used to embed *EL++* logic. This approach is very much like previous one but the main difference is that this approach tries to embed classes and their relations into boxes instead of spheres like in previous approach.

4.3.1 What knowledge base does BoxEL support

As mentioned before this embedding works with ontology written in *EL++* language.

4.3.2 How does BoxEL work

In BoxEL embedding the algorithm tries to find two functions:

$$m_w : N_C \cup N_I \to \mathbb{R}^n$$
$$M_w : N_C \cup N_I \to \mathbb{R}^n$$

First function (m_w) gives each class and individual vector representing lower left corner of box into which it will be embedded. Second function (M_w) gives each class and individual vector representing upper right corner of box into which it will be embedded.

During the train algorithm both of these functions are being trained to optimize loss function on provided training dataset.

On image 4.3 we can see how does the BoxEL embedding works on simple family onthology in two dimensional space.

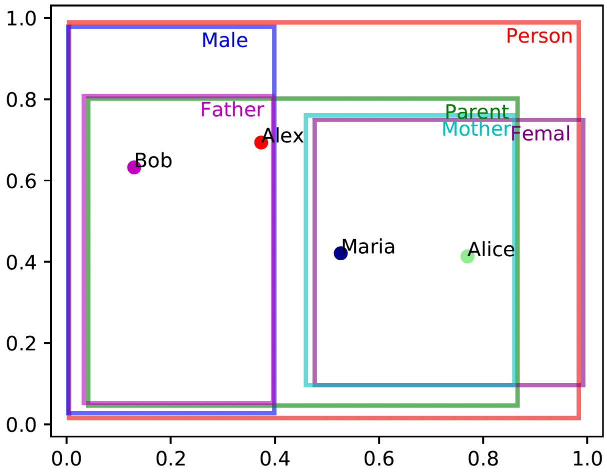

Figure 4.3: BoxEL embedding on simple family ontology [3]

4.3.3 Conclusion

In this subsection we will go trough some positives and negatives that comes with usage of BoxEL embedding in our domain.

Positives

Main positive of BoxEL embedding is that it embeds classes into separate geometric objects so that we can then also extract explanation why is file classified as malware. BoxEL embedding supports more complex class assumptions which helps us to keep as many information as possible present in final embedding. BoxEL embedding also provide advantage in expressing intersection of two classes, because intersection of two boxes is box in comparison to Sphere embedding where the intersection of two spheres can not be modeled as sphere.

Negatives

The main negative of this approach is that we can not express negation of concept. Other problem is same as it was in case of Sphere embedding - provided solution does not work.

Usability

BoxEL embedding should yield good results and should be useful for explainable malware detection.

4.4 Cone Embedding

Final approach in knowledge base embedding we will take a look at is cone embedding. This approach tries to embed classes into axis aligned cones. Axis aligned cones are convex geometrical objects. When we refer to cone we will be referring to convex cone. We can define four sets as follows: $\mathbb{R}_+ = \{x \in \mathbb{R} | x \geq 0\}$, $\mathbb{R}_- = \{x \in \mathbb{R} | x \leq 0\}$, set $\mathbb{R}$ itself and also set $\{0\}$. Then X is axis aligned cone if and only if $X = X_1 \times ... \times X_n, X_i \in \{\mathbb{R}_+, \mathbb{R}_-, \mathbb{R}, \{0\}\}$. So for example when we try to embed our dataset into 2 dimensional space there are total of 4 different axis aligned cones we can embed our data into. Another big improvement in comparison to previous embeddings in cone embedding we are able to also express negation of concept. This is allowed by way that axis aligned cones are constructed. We can define polar cone as $X^\circ = \{v \in \mathbb{R}^n | \forall w \in X :< v, w > \leq 0\}$. On image 4.4 we can see example of randomly generated vectors which form axis aligned cone and polar cone for that axis aligned cone.

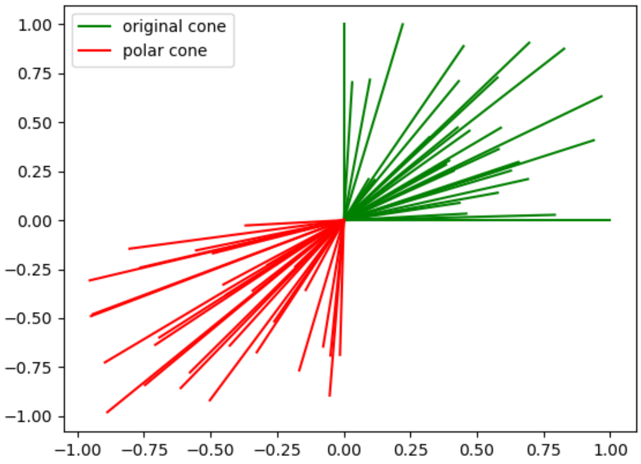

Figure 4.4: Example of axis aligned cone and it's polar cone

4.4.1 What knowledge base does Cone embedding support

Cone embedding supports ALC description logic .

4.4.2 How does Cone embedding work

Unfortunately for us the Cone embedding method is still in development and it does not have any proper implementation or algorithm which would embed provided data into vector space.

4.4.3 Conclusion

In this subsection we will go trough some positives and negatives that comes with usage of Cone embedding in our domain.

Positives

Main positive of Cone embedding is that it embeds classes into separate geometric objects so that we can then also extract explanation why is file classified as malware. Cone embedding supports more complex class assumptions which helps us to keep as many information as possible present in final embedding.

Negatives

The main negative of this approach is that the more classes ontology have the more dimensions should the target vector space have as well. So in case of some medium ontology which have around 200 classes this approach will have to embed our dataset into approximately $\mathbb{R}^{200}$ vector space.

Usability

For our research Cone embedding could probably yield good results but because of no implementation and after consultations with authors of the article this approach will probably have to be tested in future work.

Chapter 5

Proposal

Chapter 6

Implementation

Chapter 7

Results

In this chapter we will go trough results of our experiments.

7.1 Baseline

As a baseline we have decided to use TransE embedding because it is simple and easy to train. For training of TransE model we had to do some preprocessing of our dataset.

In first place we had to remove all data properties because TransE is not able to work with those. Next step was to transform our ontology dataset into RDF triples because TransE works with knowledge graphs, not ontologies. As a final step we had to split our dataset into training and testing sets. For testing set we had to remove information whether the point is malware or not so that we can then test how well can the embedding reflect this information.

7.1.1 Baseline results

When running experiments with TransE embedding we took our testing dataset and removed tail from relations that had for of individual has_type malware. Then we checked whether the prediction for that individual an relation has_type has in its top 1 choice malware or not. Based on this we evaluated testing accuracy 70.18%.

For our baseline experiments we can construct confusion matrix which would look as shown in table 7.1

TOP 1	True positives	True negatives
Predicted positives	3018	1053
Predicted negatives	1950	4048

Table 7.1: Confusion matrix for top 1 predicted tail

Chapter 8

Conclusion

Bibliography

[1] A. Bordes, N. Usunier, A. Garcia-Durán, J. Weston, and O. Yakhnenko. Translating embeddings for modeling multi-relational data. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, page 2787–2795, Red Hook, NY, USA, 2013. Curran Associates Inc.

[2] V. Gutiérrez-Basulto and S. Schockaert. From knowledge graph embedding to ontology embedding: Region based representations of relational structures. *CoRR*, abs/1805.10461, 2018.

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

[4] O. Lutfu Ozcep, M. Leemhuis, and D. Wolter. Cone semantics for logics with negation. In C. Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 1820–1826. International Joint Conferences on Artificial Intelligence Organization, 7 2020. Main track.

List of Figures