DRUG REPURPOSING WITH KNOWLEDGE GRAPHS AND MACHINE LEARNING FOR COVID-19

Pietro Spalluto, Floriana Gallo

Along with the increasing availability of drug-related data, it was necessary to find a way of handling the huge amount of information by disposing of drug knowledge bases, whose role in healthcare is becoming more relevant daily: knowledge graphs (KG) lend themselves for this role as they can easily comprehend heterogeneous data linked by a specific relation. As for the matter, different projects aimed to find a way of eventually predicting diseases' drugs; in particular, COVID-19 happened to be a priority, thus its role in drug-repurposing fed by drug-related data harvested during the pandemic. This project starts from previously implemented KG and tries to evaluate the use of two different embedding methods (TransE and DistMult) on data, relying on AUC, AUPRC and a comparison between neural network method (BPR) and standard approaches (logistic regression, random forest, XGBoost, SVC) to analyse the obtained results as a list of repurposable drugs.

I. Introduction

The aim of the project is the repurposing of previously targeted drugs to treat SARS-CoV-2 relying on Machine Learning approaches.

Several key steps are conducted from initial target identification to final clinical trials before identifying the drug as a final product, taking years of experimentation and funds with a significant risk of failure. Because of that and the increasing amount of biomedical data, the efforts are directed towards extracting useful information through ML methods: on this project, the focus was on Knowledge Graphs (KG) and their advantages. A KG is built by using nodes and labelled edges, respectively representing entities, both as heads or tails, and relations to organise them into a set of triplets: in such a manner it is possible to model complex interactions as the ones in biological systems. Traditional graphs and networks used for integrating biomedical data only contain one type of relations (e.g. interactions between proteins), while the KG provides heterogeneous information, including various entities (e.g. proteins, targets, and drugs) and multiple types of relations (e.g. interactions between drugs or drug-target pairs).

Triples are worked on through embedding techniques whose purpose is to reduce computational complexity and ease the training process as a consequence of the new low-dimensional vector spaces.

On top of the processed datasets, the Variational Graph Auto-Encoder (VGAE) method is used to further streamline the embedded data and finally outline the potentially best-performing drugs for COVID-19 treatments.

II. Related Works

The scientific scene on drug repurposing priorly focused on constructing a knowledge graph with GNN embedding methods such as Drug Repurposing Knowledge Graph (DRKG), a pretrained embedding strategy to discover an association between drugs and diseases [1]. Hsieh et al. worked on drug repurposing relying on deep graph network methods. The same was accomplished in the presented work by diversifying strategies used to organise the heterogeneous datasets available. Recent studies on the matter focus on describing KG mixed with neural network methods, for instance, Bonnet et al. [2] reviews both datasets' and approaches' choices for general drug repurposing/discovery, hence not on COVID-19 specifically.

The present work follows Hsieh et al. approach but tries to use VGAE as GNN's methods but for the embedding part, whose implementation is made with different strategies such as TransE and DistMult to compare them on the most various KG obtained. The project also takes into consideration non-neural network strategies, following Hsieh et al's paper, such as Logistic Regression, SVM, Gradient Boost and Random Forest, to evaluate if this new approach is worth the try compared to these most used methods.

III. Datasets and Features

Since the Knowledge Graph methods allow working with heterogeneous data, dataset choice was fundamental to setting the project up and proceeding with the training. In the first place, it was necessary to gather every drug-related dataset, whose choice followed two main rules: they had to be publicly available and, above all, continuously updated to obtain relevant results for the present day.

Drug-gene interactions. CTDbase provided genes and chemicals whose interaction is somewhat associated with COVID-19. The dataset itself supplies a series of information regarding the nature of the interaction, however, the main focus for the project was directed towards the solely gene-chemical interaction and their IDs to build the Knowledge Graph.

Pathways. This set of data is useful to describe known molecular interaction and reaction networks, in particular insights into molecular ones that may be affected by chemicals, other than possible mechanisms underlying environmental diseases. Inferred pathways were provided by CTDbase on top of the shared genes between its gene-disease curated list and KEGG's and REACTOME's gene-pathway curation. KEGG also gave access to PPPN pathways, processed as pairwise genes, with the interest being directed towards the PP pathway and related genes. Moreover, respiratory agents' drugs were collected from PharmGKB, whose pathways involve candidate genes. After collecting these data, gene-gene combinations were created for every pathway to consider every possible interaction.

Phenotypes. It is possible to find the Gene Ontology Consortium's (GO) ontologies on CTDbase that describe gene products in terms of their associated biological processes, cellular components, and molecular functions. Starting from that, gene-phenotype and drug-phenotype inferred associations were created for the purpose.

Host-Gene interactions. This dataset was collected from a study conducted by Gordon et al. [3] by literature search to identify proteins in human cells physically associated with the SARS-Cov-2 ones.

Trials. To finally make a ranking after training the data, listed ongoing and trial information was used to find the best-performing drugs to potentially act against contagion from Covid-19. The aforementioned dataset was obtained by literature search thanks to Hsieh et al. [4] work on the matter.

IV. Methods

The idea behind the project was to create a machine learning model able to predict the effectiveness of specific drugs to treat COVID-19 contagion. Since the information collected from data needed to be organised in a way that easily allowed relations and associations between heterogeneous elements, KG seemed the most logical outcome. Commonly used in recommender systems' applications, KG integrates expert-derived sources of information into a node-edge-node kind of graph, making a so-called triplet, where the nodes represent biomedical entities and edges represent relationships between those entities, both of them being of various natures. In this specific case, KG works with bidirectional relationships and so, for instance, a gene interacts with another gene and the interaction goes both directions. That being said, KG are being represented as a set of relations that identify edges' interactions in the term of triplet values, organised as head-relation-tail *(h, r, t)*. However, since the relationships' bidirectional nature, the distinction between heads and tails is solely conventional.

Embeddings

Once the Knowledge Graph was created starting from the collected datasets, there was the need to implement a way that could translate triplets for them to fit into the machine learning approach. Because of that, the chosen process to retain the local patterns of triplets and the global ones of the whole knowledge graph was to encode them with an embedding method [5]. The afterward prediction is performed by using the patterns to generalise the observed relationship between a specific entity and all others.

The said embedding was processed by two different models for learning low-dimensional embeddings of entities: TransE and DistMult. Both methods rely on corrupted triplets obtained with negative sampling, whose addition is necessary since KG has positive triplets only. To avoid trivial solutions, a set of corrupted triplets is made by replacing a random entity with the head or tail for every positive triplet, although not both at the same time.

TransE. As an energy-based model, TransE represents relationships as translations in the embedding space, thus its description as a translation-based model. The embedding of the tail entity t should be close to the embedding of the head entity h plus some vector that depends on the relationship r, but only if *(h, r, t)* is valid.

To learn this embedding, the loss function to minimise, whose interest is on the positive part, is a margin-based ranking criterion [6]:

$$\mathcal{L} = \sum_{(h,r,t)\in S} \sum_{(h',r,t')\in S'_{(h,r,t)}} \left[\gamma + d(h+r,t) - d(h'+r,t')\right]_{+} \tag{1}$$

where *(h, r, t)* is the positive triplet, *(h′, r, t′)* the negative one, γ is the chosen margin value; *d(h+r, t)* and *d(h′+r, t′)* represent the dissimilarity measures for positive and negative triplets, respectively. These measures d follow an energy-based framework in a way that the sum of heads and relations has to be similar to the corresponding tails' value. Thus, the negative triplet's dissimilarity measure is supposed to be higher than the corresponding positive one.

DistMult. While TransE represents the simplest additive model, with entity and relation vectors lying in the same dimensional space, DistMult is a method that relies on multiplicative operations for their vectors in $\mathbb{R}^d$ to interact. The compatibility between head and tail entities is measured with an entry-wise product as a score function [7]:

$$\sigma_{DistMult}(h,r,t) = \mathbf{r}^{\top}(\mathbf{h} \odot \mathbf{t}) \tag{2}$$

Something important to notice here is that the entry-wise product is symmetric, meaning that DistMult is not suitable for asymmetric and antisymmetric relations.

Variational Graph AutoEncoder (VGAE)

The Variational Graph AutoEncoder (VGAE) is a framework for semi-supervised learning based on Variational AutoEncoder (VAE) but specifically implemented on graph-structured data as the ones the project works on. The model's advantages lay in the use of latent variables because it is possible to obtain embedded data starting from a distribution rather than the encoding directly. Plus, the capability of creating new data in addition to the original ones enables the link prediction towards new drugs, since the VGAE accounts for the inevitable uncertainty in the knowledge graph [8].

The previously obtained embedding is used as input data and has to be encoded and decoded to provide a reconstructed output that minimises the error. Both encoder and decoder jointly go through training and optimization, so that the model can learn to compress the embedded graph structure into low-dimensional latent space in a way that ensures meaningful learned representations.

Encoder. A general framework of convolutional layers is used as an inductive node embedding and it goes under the name of GraphSAGE (SAmple and aggreGatE). It leverages node features that are incorporated into the learning algorithm, thus it is possible to learn not only the topological structure of each node's neighbourhood but even the distribution of all the node features in the neighbourhood [9]. The GraphSAGE's output is then processed with the ReLU function, dropout procedure and, after normalisation, the resulting output goes through the last two layers to compute the normal distribution through mean value μ and the logarithm of the variance value σ^2.

Decoder. The inner Product decoder is applied to a random sample from the distribution, rather than generated from the encoder directly, to reconstruct the initial input data.

Loss Function. The used loss function for VGAE has two parts: a variational lower bound and a regularizer. The first part measures how well the network reconstructs the given data so that the loss is supposed to be high with major differences between the reconstructed data and the original one; the second part, on the other hand, follows the KL-divergence, measuring the difference between P probability obtained by observed data and the probability Q obtained from the model.

Oversampling

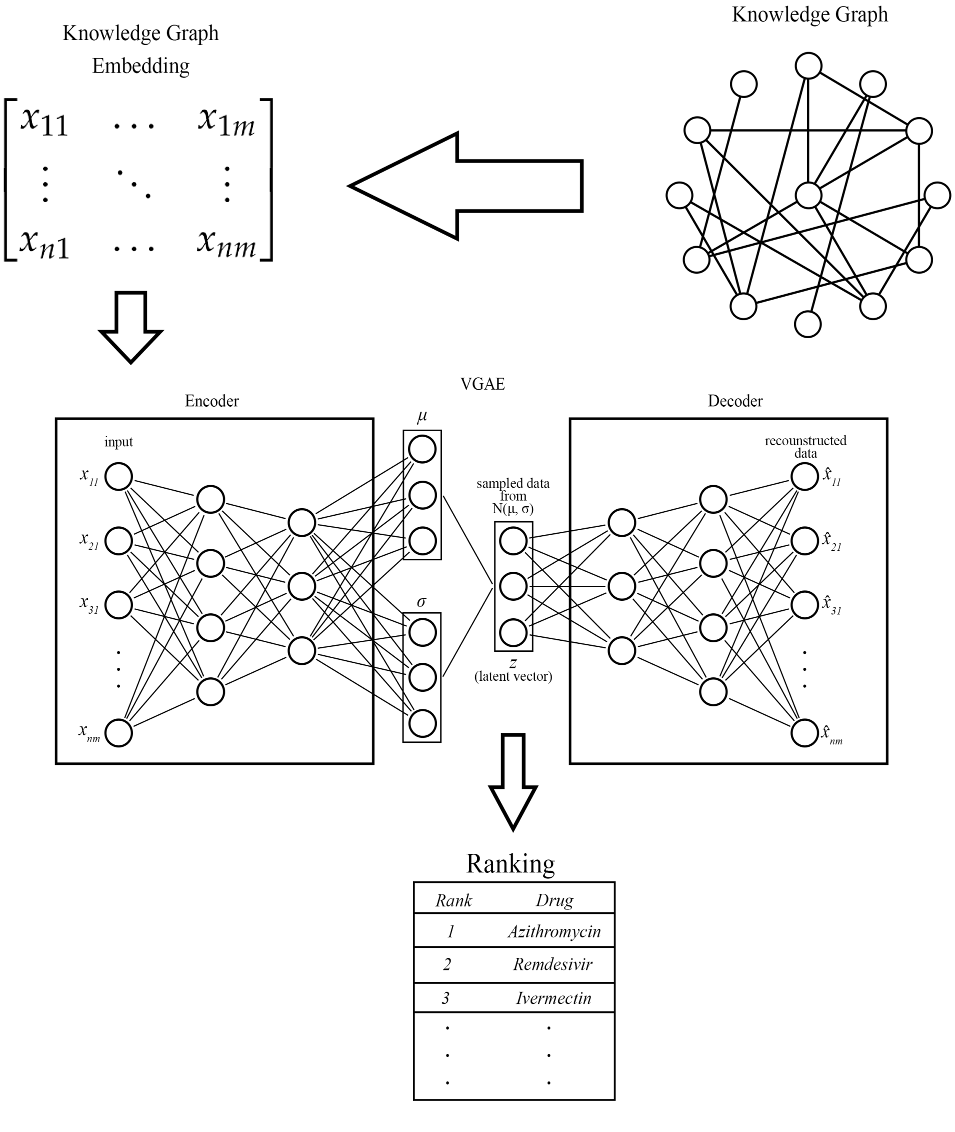
The previously listed data in the Dataset and Features section provided a final unbalanced dataset toward the negative label, making the positive one to be underrepresented: this led to questioning the actual relevance of the whole model's results since it learned how to predict negative outcomes only. To overcome this issue, an oversampling of the underrepresented data was performed using the Synthetic Minority Oversampling TEchnique (SMOTE) so that both classes could be around the same quantity.

SMOTE is a data augmentation function that creates synthetic data (oversampling) to overcome class imbalance as follows: firstly, it selects a random sample from the minority class, identifies its k nearest neighbours and then draws a vector between the current sample and one random neighbour of choice being part of the same class [10].

One important thing to notice is that the oversampling of data was performed on the training dataset only: predictions on the test set have to be realistic, hence the imbalance was necessary to preserve it as it was.

Ranking

To eventually find the best ranking drugs to treat COVID-19, the chosen method was the Bayesian Personalized Ranking (BPR), a pairwise personalised ranking loss that is derived from the maximum posterior estimator [11]. Its classifier is made of two fully connected layers whose dimension is the same as the latent space embedding in the VGAE procedure and it provides one output value representing the wanted ranking. Between the first and the second layer, ReLU, dropout and batch norm procedures are applied, whereas the second layer gives the single output as the neurons' weighted sum.



Rank	*Drug*
1	*Azithromycin*
2	*Remdesivir*
3	*Ivermectin*
.	.
.	.
.	.

Figure 1: *Schematic representation of the pipeline used. The KG is embedded in a lower dimensional space through some embedding algorithm and the resulting matrix is passed as input to the VGAE encoder. When the training process ends, the latent vector z, an encoding of the original matrix, is fed to some classification algorithms to make a ranking of the most effective drugs against COVID-19.*

V. Experiments and Results

The project involved a series of hyperparameters whose value could highly influence the model's outcome. Because of this reason, a grid search was implemented to find the combination that led to the best results. Both TransE and DistMult embedding methods were run with the same parameters and the best results were then collected to find the most reliable ranking on COVID-19's drug repurposing.

The modified parameters for the grid search are listed below:

- embedding dimension;
- batch dimension;
- learning rate;
- number of epochs;
- regularisation.

DistMult also had three different modes to compute relations' scores: normal, head batch and tail batch. Moreover, there was the chance of changing the margin for TransE.

Obtained results were saved inside a `DataFrame` and then compared by the highest value of AUC reached by both methods, run with and without oversampling to figure out its effectiveness for the case.

Table 1 summarizes for both methods related to the chosen parameters.

Table 1: *Parameters and AUC scores for both embedding methods.*

	TransE	DistMult
Embedding size	200	200
Batch size	50	80
Learning rate	0.01	0.01
Epochs	10	15
VGAE epochs	20	50
BPR classifier epochs	300	300
Regularisation	4	4.5
Margin	4	-
AUC w/o SMOTE	74%	76%
AUC with SMOTE	73%	68%

As it can be observed, best performance in terms of AUC for TransE was around 74%, while DistMult even managed to reach a 76%. Both embedding methods rose above 70% but only if not considering the oversampling of the underrepresented data, whose introduction slighly decreased results for TransE and had DistMult AUC to fall down to a 68%. The experiment comprehended a comparison between BPR classifier, a neural network based method, and other traditionally implemented methods for drug repurposing such as logistic regression, XGBoost, random forest and SVC. Results achieved with the latter methods eventually proved that the BPR classifier managed to eventually give better performance on the topic of the project rather than the usually chosen approaches for some of the combinations.

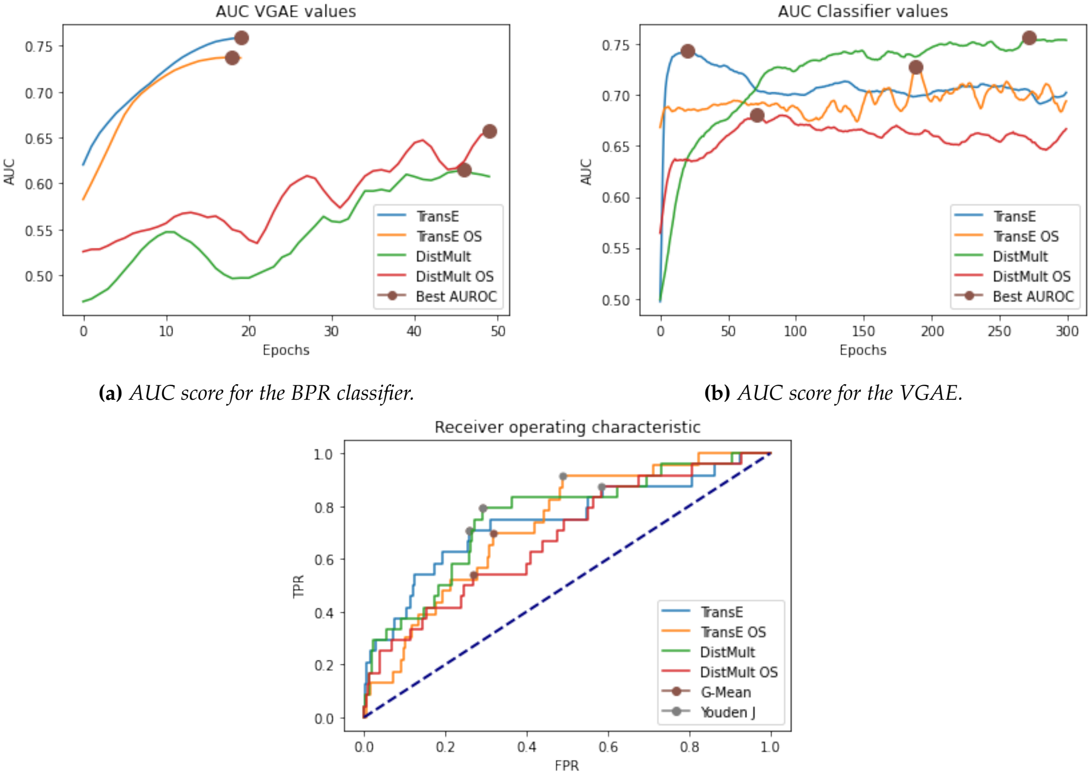
Table 2 provides results for that.

It can be noticed how some of the traditionally used strategies for drug repurposing achieved around the same AUC that was reached by the related BPR result. In particular, looking at AUCs with respect to TransE as embedding method, the oversampling executed with SMOTE relevantly decreased scores; on the other hand, it seems like traditional methods overtook BPR when oversampling was being used with DistMult.

Table 2: *AUC scores from other classification techniques.*

	Logistic Regression	Random Forest	XGBoost	SVC	BPR
TransE w/o SMOTE	**73%**	66%	**74%**	61%	**73%**
TransE with SMOTE	65%	66%	58%	64%	**74%**
DistMult w/o SMOTE	64%	56%	**76%**	70%	**76%**
DistMult with SMOTE	60%	70%	71%	72%	**68%**

Let's take a look at the AUC trend during epochs in the plots below for the four example examined and the related ROC curves, so TransE and DistMult with and without the implementation of an oversampling technique:



(a) *AUC score for the BPR classifier.*

(b) *AUC score for the VGAE.*

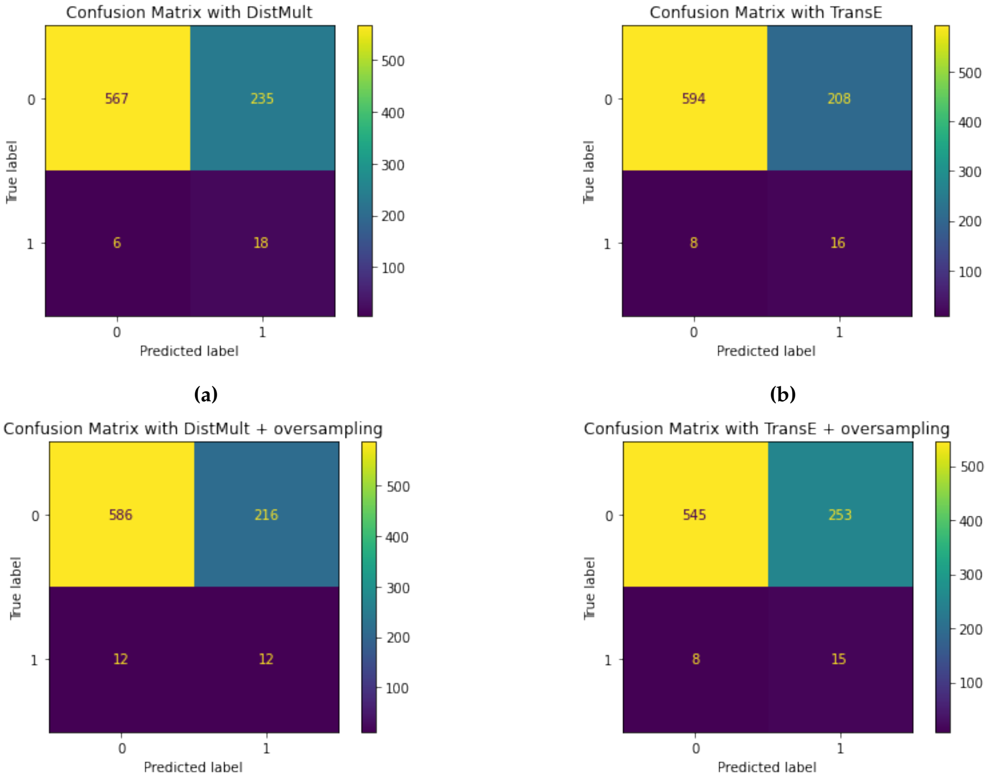
(c) *ROC curves.*

Figure 2: *Figures (a) and (b) show the AUC obtained using different embedding methods and its best value used to select the best VGAE and Classifier model. In (c) there are the 4 ROC curves relative to the best AUC values of BPR. For each curve two points are marked that represent two different thresholds (G-mean, the geometric mean of sensitivity, and Youden's J statistic) used to discriminate negative and positive values.*

As shown previously, results managed to go higher than 60% for both strategies as for the BPR. Moreover, the VGAE AUC suggests how it would be possible to obtain better results by relying on more epochs as the trend goes higher with them incrementing.

Since AUC can be affected by unbalanced data, as for this specific case with the test set having the positive class being underrepresented with a ratio of 1:35, other metrics were investigated in order to better understand the quality of the performance rather than solely relying on the AUC score.

Confusion matrix are shown below:

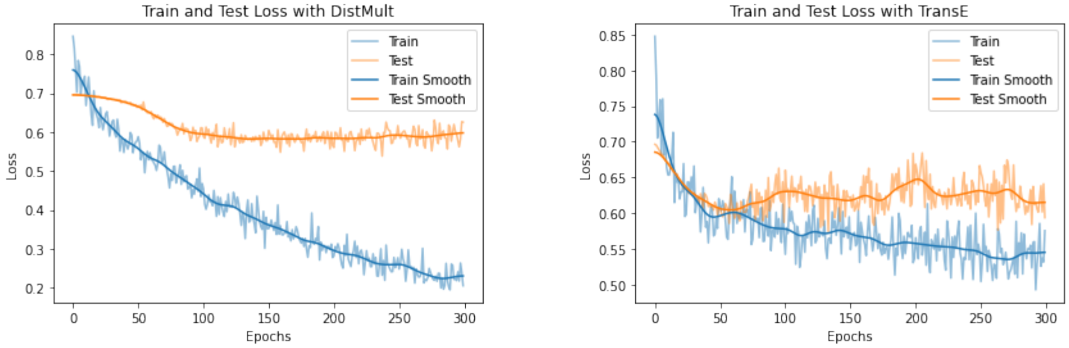


(a) **(b)**

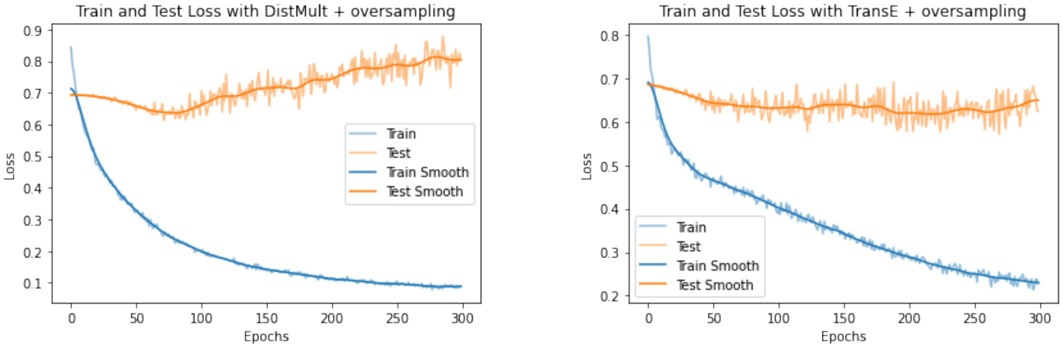
(c) **(d)**

Figure 3: *The four matrices are obtained by selecting the best classifier using the G-mean. Youden's J statistic usually gives similar results.*

Despite the effort on training the model to distinguish positive and negative classes by oversampling the underrepresented set of data, the above confusion matrix eventually turned out to affirm this lack of ability for both embedding methods, incapable of correctly predicting the negative class.



(a) **(b)**

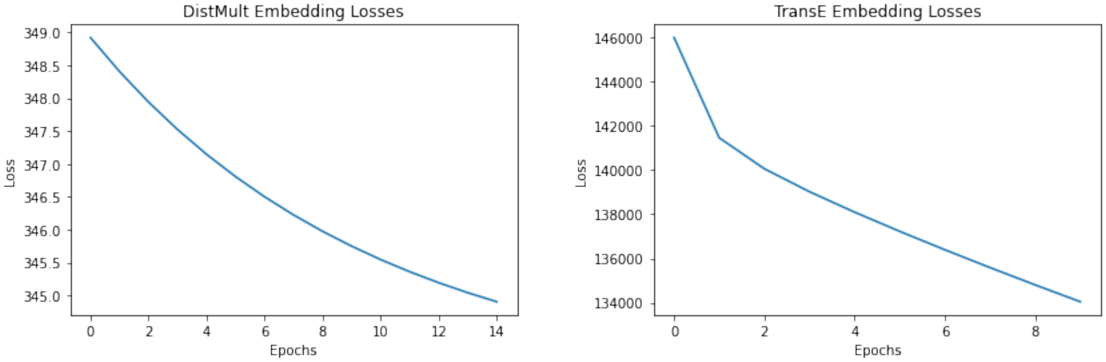


(c) **(d)**

Figure 4: *Train and test losses trends as classifier's epochs increase.*

Above are shown the losses for TransE and DistMult considering when oversampling is and is not applied; clearly, DistMult has quite a higher loss than TransE and that could be attributed to the dummy negative samples introduced in the early phase of the creation of the dataset and the oversampling when that was used. Seeing the way DistMult computes its loss, it is possible to think that a reduced amount of the wrong prediction could have "blown up" the loss since the dummy samples could be classified badly as outliers, leading to an exploding loss while the model is still capable of learning on the rest of the proposed dataset.

Lastly, a few plots showing the trend of losses for the embedding methods:



(a) **(b)**

Figure 5: *DistMult (a) and TransE (b) losses.*

VI. Conclusion

Although having achieved quite discrete results with both TransE and DistMult as embedding methods, the grid search used to find the best performance counted too many parameters to check every possible combination of values. With that being said, it can be assumed that the actual best performance lies over the already tried sets of hyperparameters in the presented work.

Both methods didn't manage to successfully predict the minority class even after oversampling the training dataset.

Moreover, even if being constantly updated to the most recent datasets, information gathered around COVID-19 and some of the interaction between pathways, drugs, genes, chemicals and so on are inferred, thus results lack the chance of giving out higher accuracy levels of prediction.

Previous work on the matter focused on finding a model able to predict associations and interactions between diseases and drugs meant to cure them, starting from standard methods to a recent use of neural network approaches, whose significantly improved results suggest how it seems to be the right pathway to follow in the field.

The present work aims to give an inside of what are the chances of a neural networks's method with diverse embedding strategies and their efficiency on drug repurposing for the specific matter of COVID-19, although it lends itself to its use in other circumstances given the right datasets. Results showed discrete use of the presented methods and they will hopefully find an improvement in the following years as diverse attempts within the scientific community are made.

References

[1] K. Hsieh, Y. Wang, L. Chen, *et al.*, "Drug repurposing for covid-19 using graph neural network and harmonizing multiple evidence," *Scientific Reports*, vol. 11, Nov. 2021. DOI: 10.1038/s41598-021-02353-5.

[2] S. Bonner, I. P. Barrett, C. Ye, *et al.*, "A review of biomedical datasets relating to drug discovery: A knowledge graph perspective," *CoRR*, vol. abs/2102.10062, 2021. arXiv: 2102.10062. [Online]. Available: https://arxiv.org/abs/2102.10062.

[3] D. E. Gordon, G. M. Jang, M. Bouhaddou, *et al.*, "A sars-cov-2-human protein-protein interaction map reveals drug targets and potential drug-repurposing," *bioRxiv*, 2020. DOI: 10.1101/2020.03.22.002386. eprint: https://www.biorxiv.org/content/early/2020/03/27/2020.03.22.002386.full.pdf. [Online]. Available: https://www.biorxiv.org/content/early/2020/03/27/2020.03.22.002386.

[4] K. Hsieh, Y. Wang, L. Chen, *et al.*, "Drug repurposing for COVID-19 using graph neural network and harmonizing multiple evidence," *Scientific Reports*, vol. 11, no. 1, Nov. 2021. DOI: 10.1038/s41598-021-02353-5. [Online]. Available: https://doi.org/10.1038%2Fs41598-021-02353-5.

[5] D. N. Nicholson and C. S. Greene, "Constructing knowledge graphs and their biomedical applications," *Computational and Structural Biotechnology Journal*, vol. 18, pp. 1414–1428, 2020, ISSN: 2001-0370. DOI: https://doi.org/10.1016/j.csbj.2020.05.017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2001037020302804.

[6] 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*, ser. NIPS'13, Lake Tahoe, Nevada: Curran Associates Inc., 2013, pp. 2787–2795.

[7] Chandrahas, A. Sharma, and P. Talukdar, "Towards understanding the geometry of knowledge graph embeddings," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 122–131. DOI: 10.18653/v1/P18-1012. [Online]. Available: https://aclanthology.org/P18-1012.

[8] T. N. Kipf and M. Welling, *Variational graph auto-encoders*, 2016. DOI: 10.48550/ARXIV.1611.07308. [Online]. Available: https://arxiv.org/abs/1611.07308.

[9] W. L. Hamilton, R. Ying, and J. Leskovec, “Inductive representation learning on large graphs,” *CoRR*, vol. abs/1706.02216, 2017. arXiv: `1706.02216`. [Online]. Available: `http://arxiv.org/abs/1706.02216`.

[10] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “Smote: Synthetic minority over-sampling technique,” *J. Artif. Int. Res.*, vol. 16, no. 1, pp. 321–357, Jun. 2002, ISSN: 1076-9757.

[11] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, *Bpr: Bayesian personalized ranking from implicit feedback*, 2012. DOI: `10.48550/ARXIV.1205.2618`. [Online]. Available: `https://arxiv.org/abs/1205.2618`.