ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

School of Information Sciences and Technology
Department of Informatics
Athens, Greece

Bachelor Thesis
in
Computer Science

Link Prediction for a COVID-19-related Knowledge Graph

Michalis Vazaios

Supervisors: Ion Androutsopoulos
Dimitris Pappas
Manolis Kyriakakis

October 2021

Michalis Vazaios

Link Prediction for a COVID-19-related Knowledge Graph

October 2021

Supervisors: Ion Androutsopoulos, Dimitris Pappas, Manolis Kyriakakis

Athens University of Economics and Business

School of Information Sciences and Technology

Department of Informatics

Natural Language Processing Group

Athens, Greece

Abstract

Knowledge graphs are networks of real-world entities where connections illustrate real-world relationships between entities. Link prediction for a knowledge graph is the procedure of finding new possible edges, which means discovering existing, but unknown at the time of the creation of the graph, relationships between pairs of the entities represented in the graph. To do this, several knowledge embedding methods have been proposed. In this thesis we firstly create a Coronaviridae-related knowledge graph with the information available at the start of the COVID-19 pandemic and afterwards we test whether existing link prediction methods can predict facts about the disease that were still unknown at the time the graph was created or that seem worth investigating to biomedical experts. To evaluate this, we show the top predictions to a half-expert (a computer scientist but with experience in biomedical topics) and conclude that link prediction methods can help add known facts that are missing from the knowledge graph and predict links that were found after the creation of the graph or that experts consider worth investigating.

Περίληψη

Οι γνωσιακοί γράφοι (Knowledge Graphs) είναι δίκτυα οντοτήτων του πραγματικού κόσμου στα οποία οι συνδέσεις αντιστοιχούν σε πραγματικές σχέσεις μεταξύ των οντοτήτων. Η πρόβλεψη ακμών για έναν γνωσιακό γράφο είναι η διαδικασία της εύρεσης νέων πιθανών ακμών, δηλαδή η ανακάλυψη υπαρκτών, αλλά άγνωστων την στιγμή που κατασκευάστηκε ο γράφος, σχέσεων μεταξύ ζευγαριών των οντοτήτων που αναπαρίστανται στον γράφο. Για το συγκεκριμένο στόχο έχουν προταθεί διάφορες μέθοδοι κωδικοποίησης/αναπαράστασης γνώσης (Knowledge Embedding). Σε αυτή την πτυχιακή εργασία αρχικά κατασκευάζουμε έναν γράφο σχετικό με τους κορωνοϊούς (Coronaviridae) με τις πληροφορίες που ήταν διαθέσιμες στην αρχή της πανδημίας COVID-19 και στη συνέχεια ελέγχουμε αν οι υπάρχουσες μέθοδοι πρόβλεψης ακμών μπορούν να προβλέψουν γνώση σχετική με την ασθένεια που δεν υπήρχε την στιγμή δημιουργίας του γράφου ή που οι βιοϊατρικοί ειδικοί θεωρούν άξια να ερευνηθεί περαιτέρω. Για να αξιολογήσουμε αυτές τις προβλέψεις, τις δείχνουμε σε έναν ημιειδικό (πληροφορικό με εμπειρία σε βιοϊατρικά θέματα) και καταλήγουμε ότι οι μέθοδοι πρόβλεψης ακμών μπορούν να προσθέσουν στο γράφο γνώση που υπήρχε την εποχή που κατασκευάστηκε αλλά που δεν είχε συμπεριληφθεί στον γράφο, γνώση που ανακαλύφθηκε μετά την κατασκευή του και πιθανές συνδέσεις που οι ειδικοί θεωρούν άξιες για περαιτέρω μελέτη.

Acknowledgements

I would like to thank everyone that has supported me with material, advice and resources during this thesis. Especially I would like to thank Ion Androutsopoulos for the continuous providing of learning resources and ideas for the thesis as well as for the opportunity he gave me to participate in valuable research. I would also like to thank Dimitris Pappas for his help with technical issues. Many thanks to Manolis Kyriakakis for helping me both with technical and more biomedical related issues and for going through the tedious task of checking the predictions of our models. Lots of thanks also to Sotiris Kotitsas for his help on how to continue his work and for his clarifications about certain points in his work. Lots of thanks also go to Causaly for providing some of the most important data for this thesis. Many thanks also to Lefteris Loukas for setting up my account to use AUEB's Kronos server. Many thanks also go to Chrysa Dikonimaki for her tips on using the Kronos server. I would also like to thank my family and friends for their heartfelt support throughout the whole procedure of this thesis.

Contents

1 Introduction

Knowledge graphs are networks of real-life entities where the connections indicate a relationship between the entities represented by the vertices. The knowledge contained in these graphs has a great variety of applications and as a result they have been widely studied. From the computational aspect of studying knowledge graphs, a point of interest is link prediction [LZ11], that is predicting new possible edges which may correspond to existing but unknown, either to the whole human kind or just to the creators of the knowledge graph, relationships between some entities of the graph.

Older models developed for this task, operate directly on the adjacency matrix of the graph and include similarity-based models [AA01] and probabilistic models [Fri+99]. More recent approaches use vector representations (embeddings) of entities and relations and achieve better results than the older approaches. Early work on knowledge embeddings includes [Bor+11], [Glo+13], [Bor+13] and others.

Examples of link prediction used in the biomedical field include predicting drug to target interactions (DTIs), where the graph entities are drugs and the proteins that the drugs interact with [WZ], drug to drug interactions (DDIs) [Kar+19] and protein to protein interactions (PPIs) [Yan+20]. In these cases link prediction is especially useful since traditionally work relies on laboratory work which is expensive and has many restrictions. Another application of link prediction in the biomedical sector is [ZAL18] where side effects of combining drugs are predicted using graph convolutional networks.

1.1 Our Contribution

We don't develop any new methods for link prediction, but instead we experiment with existing methods on a Knowledge Graph with COVID-19 related information created at the start of the pandemic. We then show our results to a half-expert and conclude that link prediction can help us discover new knowledge unavailable at the time of the creation of the graph or available but missed during the graph creation. We also get some predictions that are not established as facts in the biomedical community but that experts may consider worth investigating. This thesis continues the work of Sotiris Kotitsas MSc thesis [Kot20], where the same problem was addressed, but trying to introduce new knowledge embedding and link prediction methods that would also leverage the textual descriptors of the entities.

In this thesis, the Covid-related datasets were rebuilt from scratch, taking care to make all the experiments reproducible by explaining clearly how they were constructed. In parallel work, not reported in this thesis, methods that would also consider the textual descriptors of the entities (and relations) were also tried [Kot20], but no consistent performance gains were observed, comparing to established link prediction algorithms that do not consider textual descriptors. Hence, in this thesis we report only experiments with existing link prediction methods.

1.2 Structure of Thesis

The rest of this thesis is organised as follows:

- Chapter 2, titled "Methods", explains thoroughly the methods tried on knowledge graphs during this thesis.
- Chapter 3, titled "Datasets", describes the datasets obtained or created for this thesis. More details are provided for the creation of the Coronaviridae-related graph.
- Chapter 4, titled "Experiments", describes the experiments tried and their results. It also discusses the evaluation protocol and the results of the evaluation conducted by a biomedical half-expert.
- Chapter 5, titled "Related Work", discusses more abstractly and with less detail, other work related to ours.
- Chapter 6, titled "Conclusions and Future Work", concludes and suggests ideas for future work.

2 Methods

In this chapter we describe extensively the methods tried in this thesis and more briefly some methods we only considered for our experiments. For the methods used in our experiments we discuss the theory and the mathematics of each method while for the other methods we only provide a brief and abstract description. For a more detailed explanation of any of the methods presented, the reader should look in the corresponding papers. Before proceeding to the description of the methods, we will define the term *triplet* which will use extensively in the rest of this chapter and of the thesis. A triplet (h, r, t) signifies a relationship r between h and t where h is the subject and t is the object of the relation. For example the triplet <cyclosporine, DOWNREGULATE, genus: coronavirus> means that cyclosporine negatively regulates coronaviruses.

We adopt the categorization proposed by [Wan+b] that classifies knowledge graph embedding methods into two two classes: distance-based models and semantic matching models. The main difference between the two categories is that distance-based models exploit distance based scoring functions while semantic matching models use similarity based scoring functions. Although there is no specific reasoning of whether and, if yes, how the latter category models are more aware of the sematics of the knowledge graph, we stick with the category names proposed by [Wan+b].

2.1 Distance-based knowledge graph embeddings

Distance-based models project head and tail entities into the same vector space and then compute the distance between entity embeddings to calculate the plausibility of a given triplet. Note that the distance of two entities is different for different relations, meaning that relation embeddings are also required. Models in this category include *TransE* [Bor+13], which is the first translational distance model, its extensions *TransH* [Wan+14], *TransR* [Lin+15], *TransD* [Ji+15]. *RotatE* [Sun+19] and *OTE* [Tan+20] further extend the *TransE* method to the 2-D complex domain and to higher dimensional spaces, respectively, and are the state-of-the-art methods in this category. Below we explain more extensively the three methods of this category we used in this thesis: *TransE*, *RotatE* and *OTE*.

2.1.1 TransE

TransE embeds entities in a low-dimensional space of real-valued vectors and models relationships as translations in the embedding space: for a holding relationship (h, r, t) the embedding of the tail entity t should be close to the embedding of h plus some vector depending on the relationship r. The idea behind this translation-based model (and others that followed) was that hierarchical relationships, which are very common in Knowledge Graphs, are naturally represented by translations. For example the relation $grandchild\_of$ can easily be modelled by applying sequentially the translation for the $child\_of$ two times (the two translations applied one after the other are in fact just a new translation). Similarly, the relation $sibling\_of$ can be represented by combining the translation $parent\_of$ and the translation $child\_of$ into a new translation, which will be very close to a null translation. However, *TransE* can't model other types of relationships, present in Knowledge Graphs, such as symmetry and antisymmetry relations. These relations can also be modelled by the extensions of *TransE* explained in the following subsections (namely *RotatE* and *OTE*).

For a training set S of triplets (h, r, t), where $h, t \in E$ represent the head and tail entities respectively and $r \in R$ represents the relationship between h and t, *TransE* learns embeddings for the entities and the relationships. The embeddings e_h, e_r, e_t take values in $\mathbb{R}^k$ where k is hyperparameter of the model. Since the functional relations induced by edges of a relationship r are modelled as translations of the entity embeddings, we want $e_h + e_r \approx e_t$ to be true when (h, r, t) holds (and ideally we want e_t to be the closest neighbor of $e_h + e_r$) and we want $e_h + e_r$ to be far away from e_t otherwise. For each triplet (h, r, t) we define its energy as $d(e_h + e_r, e_t)$ where d is a dissimilarity measure: either the L_1 or the L_2 norm. To learn these embeddings the model minimizes the following hinge loss:

$$\mathcal{L} = \sum_{(h,r,t)\in S} \sum_{(h',r',t')\in S'_{(h,r,t)}} max\{0, \gamma + d(e_h + e_r, e_t) - d(e_{h'} + e_r, e_{t'})\}$$

where:

$$S'_{(h,r,t)} = \{(h', r, t)|h' \in E\} \cup \{(h, r, t')|t' \in E\}$$

is the set of all possible triplets we can get by replacing either the head or the tail of the training triplet with a random entity (this includes the head or tail of the actual training triplet) and $\gamma > 0$ is a margin hyperparameter. This means that when sampling a corrupted triplet by replacing the correct triplet's head or tail with a random entity, we want the dissimilarity score between the correct and the corrupted triplet to be at least γ.

Although *TransE* generally achieves worse results than the other methods used in this thesis, we included it as a simple and more computationally efficient version of *RotatE*.

2.1.2 RotatE

RotatE extends *TransE* by embedding entities and relationships as vectors of *complex* numbers and defining relations as rotations from the source to the target entity. *RotatE* manages to model symmetry, antisymmetry, inversion and composition relations while *TransE* is not able to model the symmetry relation. Proof about these capabilities is provided in the corresponding paper [Sun+19].

For a training set S of triplets (h, r, t), where $h, t \in E$ represent the head and tail entities respectively and $r \in R$ represents the relationship between h and t, *TransE* learns embeddings for the entities and the relationships. The embeddings e_h, e_r, e_t take values in $\mathbb{C}^k$ where k is a model hyperparameter. Since relationships are now modelled as rotations, we want $e_h \circ e_r \approx e_t$ to be true when (h, r, t) holds and we want $e_h \circ e_r$ to be far away from e_t otherwise. With $\circ$ we denote the Hadmard (element-wise) product. There is also a constraint $|e_r| = 1$ for each element of $e_r \in \mathbb{C}$, which is added to ensure that the addition of e_r indeed acts as a rotation (only changes the phases of e_h) in the complex space. Based on the above, the distance function of *RotatE* is:

$$d_r(e_h, e_t) = ||e_h \circ e_r - e_t||$$

and a loss function similar to negative loss sampling [Mik+13b] is used:

$$\mathcal{L} = -\log\ \sigma(\gamma - d_r(e_h, e_t))\ -\ \sum_{i=1}^{n} \frac{1}{k} \log\ \sigma(d_r(e_{h'_i}, e_{t'_i}) - \gamma)$$

where σ is the sigmoid function, γ is a margin hyperparameter and (h'_i, r, t'_i) is the i-th negative triplet. The intuition behind this loss function is that for a positive triplet (h, r, t), when the value $d_r(e_h, e_t)$ is much smaller than γ, then $\sigma(\gamma - d_r(e_h, e_t))$ is almost 1, resulting in the term $-\log\ \sigma(\gamma - d_r(e_h, e_t))$ becoming almost 0 (hence there is very little loss). When the value of $d_r(e_h, e_t)$ is much larger than γ, then $\sigma(\gamma - d_r(e_h, e_t))$ is almost 0, resulting in a large loss when $-\log$ function is applied. Similarly, for a negative triplet (h', r, t'), when the value of $d_r(e_{h'}, e_{t'})$ is much larger than γ, then $\sigma(d_r(e_{h'}, e_{t'}) - \gamma)$ becomes almost 1, resulting in almost 0 loss after the $-\log$ function is applied. When the value of $d_r(e_{h'}, e_{t'})$ is much smaller than γ, then $\sigma(d_r(e_{h'}, e_{t'}) - \gamma)$ is close to 0 and results in a large loss after applying the $-\log$ function.

Another approach for negative sampling has also been used with *RotatE*. This approach called self-adversarial negative sampling aims to counter the fact that some of the negative edges sampled during the training are extremely improbable. Self-adversarial negative

sampling takes negative triplets from the following probability distribution, which in effect applies a softmax to the distances $d_r(e_{h'_i}, e_{t'_i})$:

$$p((h'_j, r, t'_j) \mid \{(h_i, r, t_i)\}) = \frac{exp(-a\ d_r(e_{h'_j}, e_{t'_j}))}{\sum_i exp(-a\ d_r(e_{h'_i}, e_{t'_i}))}$$

where a is a temperature hyperparameter. Since this sampling procedure is costly, the aforementioned probability is only treated as a weight for each negative sample, but negative samples are sampled from a uniform distribution. This results in the following loss function:

$$\mathcal{L} = -log\ \sigma(\gamma - d_r(e_h, e_t))\ -\ \sum_{i=1}^{n} p(h'_i, r, t'_i)\ log\ \sigma(d_r(e_{h'_i}, e_{t'_i}) - \gamma)$$

This approach gives better results and it's the one we used in our experiments.

2.1.3 OTE

OTE extends *RotatE* from the 2D complex domain to higher dimensional spaces of real numbers and uses orthogonal transforms to model relationships. Since the rotations of *RotatE* can be viewed as orthogonal transforms in 2D complex space, *OTE* is a natural generalization of *RotatE* in higher dimensions. It maintains the ability to model symmetric, antisymmetric, inverse and compositional relationships and achieves an increased modelling capacity.

OTE uses e_h, M_r, e_t to embed each head entity, relation and tail entity (respectively). Entity embeddings $e_x \in \mathbb{R}^d$ (where d is the embedding dimension and $x \in \{h, t\}$) are each split into K sub-embeddings: $e_x = [e_x(1), ..., e_x(K)]$ where $e_x(i) \in \mathbb{R}^{d_s}$ and $d = K \cdot d_s$. Each relation embedding M_r is a set of K square matrices $M_r = [M_r(1), ..., M_r(K)]$ that serve as linear transforms for the sub-embeddings of entities (so $M_r(i) \in \mathbb{R}^{d_s \times d_s}$). [1] For each sub-embedding $e_t(i)$ of a tail entity t the projection from h and r to t is defined as:

$$\widetilde{e}_t(i) = f_i(h, r) = \phi(M_r(i))\ e_h(i) \quad (1)$$

where ϕ is the Gramm-Schmidt proccess.[2] To scale the L_2 norm of each group embedding separately, equation (1) is rewritten as:

$$\widetilde{e}_t(i) = diag(\exp(s_r(i)))\ \phi(M_r(i))\ e_h(i)$$

[1] By using sub-embeddings for entity and relation embeddings (the same way *RotatE* uses a real and an imaginary part for each component of its embeddings) a generalization of *RotatE* is achieved. This generalization results in higher modelling capacity. Note also that for $K = 2$, *OTE* becomes *RotatE* as its 2 sub-embeddings can be viewed as the real and imaginary parts of the components of *Rotate*'s embeddings.

[2] The Gramm-Schmidt proccess is employed to convert a linear transform into an orthogonal one. The advantage of using an orthogonal linear transform here is that the inverse matrix can be obtained by a simple transposition.

where $s_r(i) \in \mathbb{R}^{d_s}$ is a scalar tensor. By applying the exp function to the components of $s_r(i)$ and creating a diagonal matrix with the values of $\exp(s_r(i))$, the projection $\widetilde{e}_t(i)$ can be scaled resulting in a scaling of the L_2 norm (since the learned embeddings e_h and e_t are optimized based on the scaled per group embedding versions of $\widetilde{e}_t$ and $\widetilde{e}_h$).[3] Then, when predicting the tail of a triplet, the distance scoring function of *OTE* is defined as:

$$d(h,r,t) = \sum_{i=1}^{K}(||\widetilde{e}_t(i) - e_t(i)||)$$

Similarly, when predicting the head h of a triplet, the projection of a sub-embedding $e_h(i)$ from r and t to h is defined as:

$$\widetilde{e}_h(i) = diag(exp(-s_r(i)))\ \phi(M_r(i))^T\ e_t(i)$$

and the distance scoring function as:

$$d(h,r,t) = \sum_{i=1}^{K}(||\widetilde{e}_h(i) - e_h(i)||)$$

The loss function used is the same as in *RotatE*. *OTE* also uses the same self-adversarial strategy for negative sampling, and that was the approach we used in our experiments as well, since it produces better results.

2.2 Semantic Matching Models

Semantic matching models use multiplicative score functions to compute the probability of any given triple. Examples of such models include *DistMult* [Yan+15], *ComplEx* [Tru+16], *ConvE* [MSR18], *QuatE* [Zha+19] and *TuckER* [BAH19a]. We use the last one for our predictions.

2.2.1 TuckER

TuckER uses Tucker Decomposition [Tuc64] on the binary third-order tensor representation of the knowledge graph. For a knowledge graph with $|V|$ number of entities (vertices) and $|R|$ relations, the binary tensor representation of the graph is $\mathcal{X} \in \{0,1\}^{|V|\times|R|\times|V|}$ where $\mathcal{X}[h,r,t] = 1$ if there exists a triplet (h,r,t) and $\mathcal{X}[h,r,t] = 0$ otherwise. The Tucker Decomposition of $\mathcal{X}$ is:

$$\mathcal{X} \approx \mathcal{Z} \times_1 E \times_2 R \times_3 E'$$

[3] A version of *OTE* without the diagonal scalar tensor was tried in the original paper, giving slightly worse results than the scaled version.

where $E \in \mathbb{R}^{|V| \times d_e}$ and $E' \in \mathbb{R}^{|V| \times d'_e}$ contain the embeddings for the head and tail entities respectively, $R \in \mathbb{R}^{|R| \times d_r}$ contains the embeddings for the relations of the graph and $\times_n$ denotes the tensor product along the n-th axis of two tensors. d_e, d_r, d'_e are the dimensionalities of the embeddings, and $\mathcal{Z} \in \mathbb{R}^{d_e \times d_r \times d'_e}$ is the core tensor of the result of the Tucker Decomposition. The matrices E and E' can be set to be equal so that each entity has the same embedding when it appears as head or tail.

The scoring function of the model is:

$$\phi(h, r, t) = \mathcal{Z} \times_1 e_h \times_2 e_r \times_3 e_t$$

where e_h, e_r, e_t are the embeddings of h, r, t respectively. The logistic sigmoid function (or another sigmoid function) is applied to these scores to get the probability of each triplet.

The embedding matrices E, R and the core tensor $\mathcal{Z}$ are learned via backpropagation by trying to minimize a Bernoulli negative log-likelihood loss function. The loss function for each incomplete triplet is defined as the sum of the binary cross-entropies of all the entities that could be used to fill the missing entity of the triplet:

$$\mathcal{L} = -\frac{1}{|V|} \sum_{i=1}^{|V|} (y^{(i)} log(p^{(i)}) + (1 - y^{(i)}) log(1 - (p^{(i)})))$$

where $p \in (0, 1)^{|V|}$ is the vector of predicted probabilities and $y \in \{0, 1\}^{|V|}$ is the binary label vector.

TuckER is a fully expressive model, that is for any ground truth it can find entity and relation embeddings that separate the true from the false triplets. The proof for this includes setting a bound to the embedding dimensions of the entities and relations. In practise however, the embeddings used are much smaller since the values of the binary tensor of the graph are not random but have a certain structure (which is what enables the model to predict new entities). Moreover, lower dimensions for the embeddings are desirable since they allow the model to make better generalizations, instead of memorizing its training input.

2.2.2 Other models considered

DistMult uses real-valued vectors to embed each entity and a linear equation to embed the connection between the entities through a relation. It uses a symmetric score function (with the same scores for triples <h, r, t> and <t, r, h>) and as a result it cannot model asymmetric relationships for which only one direction is valid (e.g. X, father_of, Y).

ComplEx is an extension of *DistMult* that uses complex-valued representations of entities and that is able to model asymmetric relationships as well.

QuatE is in turn a generalization of *ComplEx*. It uses hypercomplex-valued embeddings with three imaginary components to represent entities, resulting in a better modelling capacity.

ConvE uses multiple fully-connected 2D convolution layers to model interactions between entities and relationships.

3 Datasets

In this chapter we firstly discuss the two benchmark datasets we used to ensure we can replicate the previously reported performance of established link prediction algorithms and to measure their computational efficiency; the latter is often neglected in previous work. Then we describe the Coronaviridae-related graph, which we created ourselves to predict unknown interactions with the disease COVID-19 and with the virus causing it, SARS-CoV-2. On Table 3.1 we provide basic statistics about the benchmark datasets and on Table 3.2 we do the same for the Coronaviridae-related graph.

3.1 WN18RR

The first benchmark dataset, WN18RR, is derived from WordNet [Fel98]. WordNet is a database of lexical semantic relations, such as hypernym-hyponym (broader-narrower concept), synonyms, meronyms (part-of) etc. WN18 was originally created by Bordes et al. [Bor+13], and was later improved to its current version WN18RR by Dettmers et al. [Det+18], who also ensured that the development and test subsets do not include inverses of training facts (e.g., $\langle A, \text{is-hypernym-of}, B\rangle$ vs. $\langle B, \text{is-hyponym-of}, A\rangle$), which would be trivial to predict. As a result, most link prediction methods perform much better on WN18 than on WN18RR.

3.2 FB15K-237

The second benchmark dataset, FB15K-237, is a subset of Freebase.[1] Nodes and edges capture facts about movies, actors, sport teams, players etc. The dataset, then called FB15K, was originally created by Bordes et al. [Bor+13], but was improved by Toutanova et al. [TC15]. The main improvements were again related to ensuring that the development and test subsets do not include inverses of training facts. Relation types were also reduced from 1,345 to 237. As with WN18 and WN18RR, most link prediction methods give better results for FB15K than for FB15K-237 as a result of the removal of the trivial to predict edges.

[1] en.wikipedia.org/wiki/Freebase_(database)

Statistics	WN18RR	FB15K-237
Nodes	40,943	14,541
Edges	93,003	310,116
Training edges	86,835	272,115
Development edges	3,034	17,535
Test edges	3,134	20,466
Relationship types	11	237

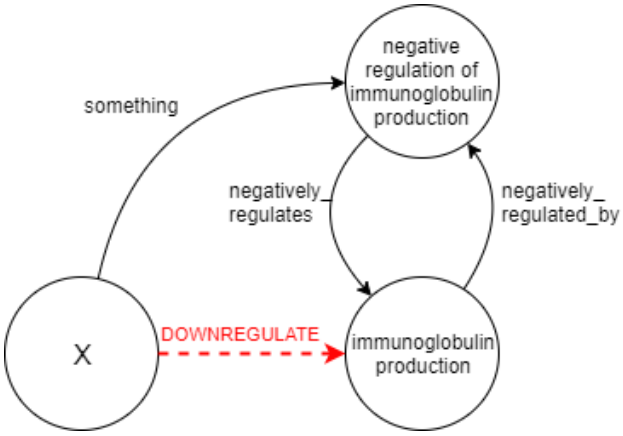
Tab. 3.1: Statistics of WN18RR and FB15K-237.

3.3 COVID-19-Graph

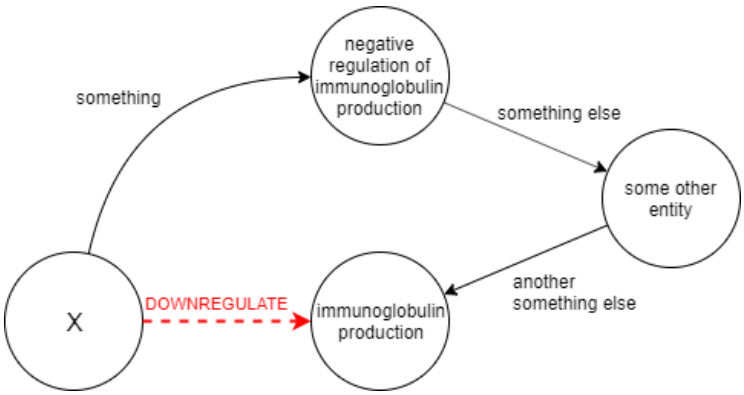
For our main experiments, we firstly created a new knowledge graph about the Coronaviridae, the family of viruses that includes SARS-CoV-2, the virus causing COVID-19. This graph, hereafter called CORONA-GRAPH, is an extension of an initial graph kindly provided to us by Causaly[2]. The initial graph was constructed in May 2020, using automatic information extraction from biomedical literature (mostly journal articles) and subsequent manual curation by biomedical experts. Subsets of that graph were provided to interested parties, including ourselves. Each edge of the graph we were given represents a known (at the time the graph was constructed) interaction between a biomedical entity (e.g., drug, gene) and (i) a virus (or group of viruses) of the Coronaviridae or (ii) a disease caused by a virus of that family. There were 816 nodes, 1,229 edges, and 4 relationship types (edge labels): UPREGULATE, DOWNREGULATE, CAUSES-REACTION, INTERACTS-WITH[3]. The first two relationship types indicate a positive or negative reaction, respectively, from the source to the target of the edge (e.g., <cyclosporine, DOWNREGULATE, genus_coronavirus>). CAUSES-REACTION signals that the source entity causes a reaction on the target, but it is not established if the reaction is positive or negative (e.g., due to conflicting reports). INTERACTS-WITH indicates the entities interact, but both the direction and polarity of the reaction are not established.

To include broader biomedical knowledge in the graph (e.g., <cyclosporine, is_a, antifungal_agent>), for each entity of the initial graph we added the entity's neighbors from the UMLS ontology [4] up to two hops away, along with the edges corresponding to the hops. Before performing the hops on the UMLS graph we removed edges with labels 'regulates', 'regulated_by', 'negatively_regulates', 'negatively_regulated_by', 'positively_regulates', 'positively_regulated_by'. This was done to avoid predicting trivial edges like the red one in the image below.

[2] `www.causaly.com`
[3] In the original graph and in Sotiris' thesis the latter two were called UNIDIRECTIONAL and BIDIRECTIONAL respectively.
[4] `www.nlm.nih.gov/research/umls`



However, since the tail entities for relations such as 'negatively_regulates' contained in their names the concept of regulation, nodes like 'negative regulation of immunoglobulin production' needed to be excluded as well to avoid making easy predictions like the one demonstrated below with red:



Here it could be argued that it would be better to just remove nodes like 'negative regulation of immunoglobulin production' and keep edges like the red DOWNREGULATE one of the images above. We did not follow this approach because it would require a rule-based method of adding DOWNREGULATE, UPREGULATE and CAUSES-REACTION edges based on existing edges. This rule-based method could be something similar to "If we have the triplets <A, something, negative regulation of B> and <negative regulation of B, negatively_regulates, B>, then add to the knowledge graph a DOWNREGULATE edge from A to B". However, any such approach would have to be devised by the writer of this thesis (who is **not** a biomedical expert) and could result in adding many non manually curated edges to the graph.

In addition to this pruning of the UMLS graph, to keep the resulting graph at a manageable size, we used hops corresponding only to the 28 most frequent relationship types of UMLS, out of the 916 total. After expanding the graph, we also removed UMLS edges of relationship types occurring fewer than 200 times in the expanded graph, ensuring that

the graph remained connected. During this last part of the edge removal the graph is split into two (weakly) connected components, one with 315225 entities and one with 3 entities. We keep the former.

Finally, we added nodes and edges from CHEMBL, a manually curated database of bioactive molecules with drug-like properties. [5] CHEMBL provides 'drug indication' and 'mechanism of action' interactions. For every entity v of our graph, if CHEMBL provided a 'drug indication' interaction between v and a disease, we added the interaction to the graph as a DOWNREGULATE edge, since in this case v is a drug for the disease. Also, if CHEMBL provided a 'mechanism of action' for v, we added the interaction to the graph by mapping the action type to one of the four relationship types of the initial graph (see table 7.1 in the Appendix for the mapping). During this procedure, nodes that didn't exist in the previous version of our graph were added, but only as target tail nodes. The reason for adding nodes was that if we added only edges having both endpoints in the existing nodes of the graph, then we wouldn't add enough new nodes in this phase (there were 1319 edges in CHEMBL with both endpoints in the existing set of nodes). So we needed to add nodes in one of the following ways: only as source nodes, only as target nodes, both as target and as source nodes. There were 1741 edges in CHEMBL with their target node in the existing set of nodes, 3725 edges with their source node in the existing set of nodes and 4147 edges with either their source node or their target node in the existing set of nodes. As a result, adding nodes only as target nodes would result in a sufficient amount of edges to use as training, development and testing triplets.[6] Another reason for choosing to add nodes only as target ones was to be more consistent with the work of Sotiris Kotitsas [Kot20] who created a Covid-related graph in a similar way and only added tail nodes when he used CHEMBL to enrich his graph.

At this point we had created the CORONA-GRAPH, which we used for evaluating the models on a graph about the Coronaviridae, in experiments similar to those we performed using WN18RR and FB15K-237. The CORONA-GRAPH however contained neither the entity for SARS-CoV-2 (the virus that causes COVID-19) nor the COVID-19 disease itself, since it was created based on the 2019 version of UMLS, which did not contain those two entities yet. Since we eventually wanted to predict interactions between SARS-CoV-2 and other biomedical entities to show to experts, we added to CORONA-GRAPH triplets of the four relationship types UPREGULATE, DOWNREGULATE, CAUSES-REACTION, INTERACTS-WITH that contained the entity 'coronavirus cov-19' as either target or source, creating the final version of our graph, the COVID-19-Graph. These additional triplets were obtained from another, non-curated graph, also provided to us by Causaly (constructed in May 2020), which included an entity 'coronavirus cov-19' denoting both SARS-CoV-2 and COVID-19. The additional triplets contained noisy and sometimes very

[5] `www.ebi.ac.uk/chembl`

[6] The edges added in this stage were 3697 because some of the 3725 edges with (at least) their source nodes in the existing set of nodes also existed in the original graph provided by Causaly

generic information about SARS-CoV-2, since they were not manually curated by experts and because the bibliography relative to SARS-CoV-2 was limited at the time they were extracted from biomedical literature.

Original graph on Coronaviridae	
Nodes	816
Edges	1,229
Relationship types	4
With UMLS edges added	
Nodes	315225
Edges	839306
Relationship types	27
With CHEMBL edges added (CORONA-GRAPH)	
Nodes	315,739
Edges	843,002
Relationship types	27
Predicted relationship types (of interest)	4
Training edges (all types)	842,402
Training edges (4 predicted types)	4,326
Development edges (4 predicted types)	300
Test edges (4 predicted types)	300
With COVID-19 edges added (COVID-19-Graph)	
Nodes (including 'coronavirus cov-19')	315,740
Edges (all used for training)	843705
Relationship types	27
Predicted relationship types (of interest)	4
Edges (4 predicted types, all used for training)	703

Tab. 3.2: Statistics of the datasets used in our main Covid-related experiments. Note that even though the models are trained to predict relations of all 27 relationship types, we evaluate each model on its ability to predict only the relations of the 4 types of interest (UPREGULATE, DOWNREGULATE, CAUSES-REACTION, INTERACTS-WITH)

4 Experiments

In this chapter we firstly define the evaluation measures used for our experiments and then we present our experimental results . We present our results both on development and test data for each model and each dataset. Afterwards, we present the evaluation protocol used by the half-expert to assess our predictions and we discuss the results of the evaluation.

4.1 Evaluation measures

To measure the performance of the models tried in our experiments we employed three evaluation metrics: MRR, MR and Hits@k. These metrics can be used for evaluating any process that ranks possible responses to a sample of queries by probability of correctness. The task of link prediction is just a specific case of such a process, making the use of these metrics natural for the evaluation of models.

MRR stands for **M**ean **R**eciprocal **R**ank. The reciprocal rank of the response to a query is the inverse of the rank of the correct answer: 1 for first place, $\frac{1}{2}$ for second place, and so on. The **M**ean **R**eciprocal **R**ank for a set of queries Q is defined as the mean of the reciprocal ranks of the results of these queries:

$$MRR = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank_i}$$

where $|Q|$ is the number of queries and $rank_i$ is the rank position of the correct answer for the i-th query.

MR stands for **M**ean **R**ank and it is simply the mean of the ranks of the correct answers:

$$MR = \frac{1}{|Q|}\sum_{i=1}^{|Q|}rank_i$$

Hits@k refers to the percentage of the queries for which the correct answer had a rank of at most k. We calculate Hits@k for k $= 1, 3, 10$ during the evaluation process.

4.2 Experiments on benchmark datasets

On Table 4.1 we can see the results of the four methods on test data and on Table 4.2 the results on development data. We also specify the training times for each model. Only the training time is reported, since the inference time is negligible compared to the training time. On Table 4.1, we also reference the results others achieved in the WN18RR and FB15K-237 datasets. Note that we cite the paper achieving the mentioned results and not necessarily the paper introducing the corresponding model (e.g. when *TransE* was created, the two datasets didn't exist).

For the WN18RR and FB15K-237 datasets, all the algorithms ran on an NVIDIA GTX 1080 8GB GPU, except *OTE* which ran on Google Colaboratory (Colab) on a Tesla P100-PCIE-16GB GPU. For the CORONA-GRAPH dataset all the models ran on Colab on the same GPU. We didn't run *TransE* on CORONA-GRAPH because of its poor results in the previous two datasets.

Our results were generally similar to previous work. We notice that *TransE* is outperformed by *RotatE*, which in turn is outperformed by *OTE*. This was expected since *RotatE* is an extension of *TransE* and *OTE* is an extension of *OTE*. *RotatE*, *OTE* and *TuckER* produce similar results in the two benchmark datasets and while *TransE* gives far worse results. On CORONA-GRAPH the differences between the results of the models tried are bigger but this can be possibly attributed to the fact that, contrary to the case with the two benchmark graphs, no previous work on which we could step on was done. We also notice that *TuckER* produces better results than *OTE* (the best out of the three distance-based models) on FB15K-237 and CORONA-GRAPH while it is outperformed both by *RotatE* and *OTE* on WN18RR. We can see that WN18RR is an 'easier' graph than FB15K-237 since models tend to perform better on it. This can be attributed to the lower number of relationship types in it. CORONA-GRAPH has less relationship types than FB15K-237 however the models tried on CORONA-GRAPH, except *TuckER*, tend to go worse on it than on FB15K-237. A possible explanation for this is the fact that the model is trained on 27 relationship types but evaluated only on 4 types of edges and that *TuckER*'s similarity-based approach is better at leveraging information from all 27 relationship types than the other models' distance-based approach. Regarding the training times for the models we do not notice any particular pattern. Although some models may be theoretically more computationally efficient than others, the actual measure for their efficiency is the time they need to converge, which cannot be predicted since it depends on the dataset and the model hyperparameters.

WN18RR	**MRR**	**MR**	**Hits@k** (%)			**Training Time**
	(%)		@1	@3	@10	(hours)
RotatE (our results)	47.2	3394.4	42.4	48.9	57.1	2.52
RotatE [Sun+19]	47.6	3340	42.8	49.2	57.1	n/a
TransE (our results)	23.0	3200.2	2.4	40.2	52.8	2.52
TransE [Ngu+18]	22.6	3384	n/a	n/a	50.1	n/a
TuckER (our results)	46.6	6428.9	43.7	48.0	52.2	5.67
TuckER [BAH19a]	47.0	n/a	44.3	48.2	52.6	n/a
OTE (our results)	48.3	2973.9	43.3	50.1	58.5	2.62
OTE [Tan+20]	48.5	n/a	43.7	50.2	58.7	n/a
FB15K-237	**MRR**	**MR**	**Hits@k** (%)			**Training Time**
	(%)		@1	@3	@10	(hours)
RotatE (our results)	33.3	172.3	23.8	36.9	52.5	3.29
RotatE [Sun+19]	33.8	177	24.1	37.5	53.3	n/a
TransE (our results)	29.5	214.8	20.1	33.0	47.7	1.06
TransE [Ngu+18]	29.4	347	n/a	n/a	46.5	n/a
TuckER (our results)	35.5	163.2	26.4	38.9	53.6	4.20
TuckER [BAH19a]	35.8	n/a	26.6	39.4	54.4	n/a
OTE (our results)	35.2	174.3	25.9	38.7	53.8	3.51
OTE [Tan+20]	35.1	n/a	25.8	38.8	53.7	n/a
CORONA-GRAPH	**MRR**	**MR**	**Hits@k** (%)			**Training Time**
GRAPH	(%)		@1	@3	@10	(hours)
RotatE	24.4	3404.1	16.0	26.3	44.5	5.68
TuckER	36.9	10002.1	32.7	37.5	46.7	3.71
OTE	30.4	2141.2	22.2	33.0	48.8	2.18

Tab. 4.1: Results on **test** data. Non-referenced results come from our own experiments.

WN18RR	MRR (%)	MR	Hits@k (%) @1	@3	@10
RotatE	47.6	3340.7	43.1	48.8	56.6
TransE	22.6	3177.8	2.1	40.2	52.5
TuckER	46.3	6143.7	43.6	47.4	51.1
OTE	48.9	2887.8	44.3	50.3	58.3

FB15K-237	MRR (%)	MR	Hits@k (%) @1	@3	@10
RotatE	33.8	159.6	24.3	37.2	52.8
TransE	29.9	203.7	20.8	33.3	48.1
TuckER	36.0	150.6	27.1	39.1	54.0
OTE	35.9	160.4	26.7	39.4	54.2

CORONA-GRAPH GRAPH	MRR (%)	MR	Hits@k (%) @1	@3	@10
RotatE	24.7	4420.1	16.8	28.5	39.2
TuckER	33.9	9984.2	29.3	35.3	42.3
OTE	29.2	2411.5	20.8	32.0	45.0

Tab. 4.2: Results on **development** data. All the results are from our own experiments.

4.3 Experiments on final COVID-19-Graph

After benchmarking the models on CORONA-GRAPH, we chose *TuckER*, based on its MRR and Hits@k metrics to make our final predicitions of interactions between SARS-CoV-2/COVID-19 and other biomedical entities on the COVID-19-Graph. Since we are interested only in interactions between SARS-CoV-2/COVID-19 and certain types of biomedical entities and to avoid noisy predictions, we restrict our predictions to include only these categories of entities (see Appendix 7.2). We take the 100 top-scored predictions for each of the following 8 cases:

- <Something, DOWNREGULATE, SARS-CoV-2/COVID-19>
- <Something, UPREGULATE, SARS-CoV-2/COVID-19>
- <Something, CAUSES-REACTION, SARS-CoV-2/COVID-19>
- <Something, INTERACTS-WITH, SARS-CoV-2/COVID-19>
- <SARS-CoV-2/COVID-19, DOWNREGULATE, Something>
- <SARS-CoV-2/COVID-19, UPREGULATE, Something>
- <SARS-CoV-2/COVID-19, CAUSES-REACTION, Something>
- <SARS-CoV-2/COVID-19, INTERACTS-WITH, Something>

and keep the predictions with probability score ≥ 0.05 to show to a half-expert. We end up with 91 predictions in total (see Table 7.2 in Appendix). Out of these, 80 are of the type <Something, DOWNREGULATE, SARS-CoV-2/COVID-19>, which can be considered possible treatments or ingredients of possible drugs for COVID-19 disease. Another 7 of the 91 predictions that pass the 0.05 treshold are of the form <SARS-CoV-2/COVID-19,

UPREGULATE, Something>. These relationships could indicate symptoms of the disease or help find ways to test whether someone is a carrier of the disease or not. The other 4 predictions are of the INTERACTS-WITH relations with SARS-CoV-2/COVID-19, which are less usefull and more generic than the previous two types of predictions.

4.4 Half-expert evaluation

4.4.1 Evaluation protocol

We gave our biomedical half-expert a list of pairs coming from the 91 final predictions of the previous section. Each pair was of the form (Entity 1, Entity 2), where the first of the two entities is COVID-19/SARS-CoV-2,[1] and the other entity of the pair is another biomedical entity (e.g., a gene or chemical compound). Since the evaluation protocol does not take into account neither the type of predicted relationship nor the direction of the relation, the 91 predictions were reduced to 88 pairs because of entities for which the model predicted that they are related to COVID-19/SARS-CoV-2 with more than one of the following four relationship types. Each pair denotes an automatically generated prediction that there is a relationship of any of the following four types between Entity 1 and Entity 2. However, the half-expert was not told the predicted relationship type of each pair. For his purposes, a pair was correct if and only if there existed sufficient evidence (in biomedical articles or databases) that Entity 1 was linked to the Entity 2 with *any* of the following four relationship types.

UPREGULATE: There is a positive reaction from one of the two entities to the other.

DOWNREGULATE: There is a negative reaction from one of the two entities to the other.

CAUSES-REACTION: One of the two entities causes a reaction to the other (and we know the direction of the reaction), but it is not established if the reaction is positive or negative.

INTERACTS-WITH: The two entities interact, but both the direction and the polarity (positive or negative) of the reaction are not established.

For each pair, we asked our half-expert to search for evidence (in biomedical articles or databases) supporting or rejecting the pair's correctness and answer the following

[1] Remember that we use a single entity to represent both the virus SARS-CoV-2 and the disease COVID-19 (see Section 4.3)

questions. The questions we asked him and the options we gave him are presented below:

Question 1: Based on the *present* evidence (September 2021), which of the following best describes the (Entity 1, Entity 2) pair? Select one option.

5 – The pair is correct. Concrete supporting evidence found. Please also tell us the evidence and the (single) correct relationship type (up-regulates, down-regulates, causes-reaction, interacts with) based on the evidence.

4 – The pair is probably correct. Some supporting evidence found, but it is not a fully established finding yet. Please also tell us the evidence and the (single) most likely relationship type based on the evidence.

3 – The pair may or may not be correct, but it is worth investigating. There is not enough evidence to tell if the pair is probably correct or not, but there is evidence suggesting it is worth investigating the Entity 1-Entity 2 interaction in further research. Please also tell us the evidence and your best guess for the (single) relationship type based on the evidence.

2 – The pair may or may not be correct, but is *not* worth investigating. There is not enough evidence to tell if the pair is probably correct or not, and it does *not* seem worth investigating the Entity 1-Entity 2 interaction further. Please also tell us any related evidence you found, if any at all.

1 – The pair is probably incorrect. No supporting evidence found or some evidence against it found, though the pair cannot be rejected with certainty.

0 – The pair is incorrect. Concrete evidence against it found. Please also tell us the evidence.

Question 2: The predictions of the 91 triplets are based on information collected at the beginning of the COVID-19 pandemic. You will be told exactly when the information was collected (May 2020). Please repeat Question 1, but now using evidence available *at the time the information the predictions are based on was collected.* If your answers to Questions 1 and 2 are different for a pair, please also provide brief comments to help us understand the reasons for the differences (e.g., an Entity 1-Entity 2 link that seemed worth investigating at the beginning of the pandemic has now been investigated and we now know it is not worth considering further).

4.4.2 Results

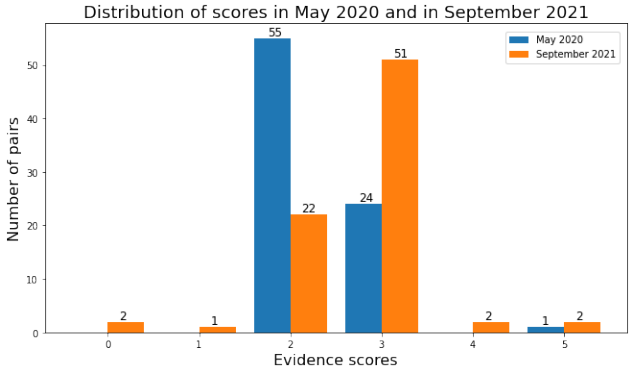
Our biomedical half-expert was kind enough to include other types of answers apart from assessing the correctness of each triplet with a number in the range 0-5. Specifically, he pointed out that 6 of our predicted relations were very generic information e.g. 'DNA-binding proteins is a too generic topic. Of course SARS-CoV-2 interacts with many DNA-binding proteins'. He also categorised 2 of our predictions as noise, that is predictions of relations between COVID-19/SARS-CoV-2 and a completely irrelevant entity that wouldn't have any meaning connecting to COVID-19/SARS-CoV-2 (e.g. high occupancy target region). For the full list of the evaluation of pairs by the half-expert see Appendix 7.3.

For the remaining 80 pairs the results were mostly mediocre (scores 2 or 3). Table 5.1 shows the count of predictions for each combination of correctness score in May 2020 and September 2021.

Evidence score May 2020	Evidence score September 2021	Count of predictions
2	2	22
2	3	33
3	0	2
3	1	1
3	3	18
3	4	2
3	5	1
5	5	1

Tab. 4.3: Count of predictions by evidence score in May 2020 and September 2021

The plausibility of most pairs had either stayed the same or slightly increased from May 2020 to September 2021. The bar chart below shows the distribution of the scores in May 2020 and September 2021.



Although both in May 2020 and September 2021 the majority of scores for the pairs is either 2 or 3, we see that in September 2021 the most usual score for the predicted pairs is

3 (pair that may or may not be true but is worth investigating) while in May 2020 the most usual score is 2 (pair that may or may not be true but is *not* worth investigating).

We notice that there exists a case where the model discovered an interaction we now know to be correct (evidence score 5) which the semi-expert only considered worth investigating (evidence score 3) based on what was known in May 2020: this case concerned rituximab; in September 2021 it was known with certainty that treatment with rituximab is a risk factor for severe and prolonged COVID-19 infection [Yas+20], but this was only a hypothesis worth exploring in May 2020. However, in this case the prediction of our model was <COVID-19/SARS-CoV-2, UPREGULATE, rituximab>, which means that the model found the correct relation but with the wrong direction. We also note a case where the score of the predicted pair was 5 both in May 2020 and in September 2021. This means that the model found an interaction known in May 2020 but not included in the initial Causaly graph. This prediction concerned a DOWNREGULATE relation from morphine sulphate to COVID-19/SARS-CoV-2.

There exist 3 cases where the model predicts interactions that seemed worth investigating at the start of the pandemic but we now know to be incorrect or probably incorrect. The two cases of entities that seemed worth investigating at the start of the pandemic but that we now know to be incorrect were chloroquine hydrochloride and chloroquine sulfate. Both are chloroquine derivatives and although chloroquine and its derivatives seemed as potential effective treatments for COVID-19 in the beginning of the pandemic, subsequent trials proved that chloroquine has no real effect against COVID-19. The third case concerns amodiaquine which is an anti-malarial drug with a mollecular structure very close to chloroquine, so it was also considered as a possible treatment against COVID-19 at the start of the pandemic and it produced promising results in experiments on hamsters. However, given the experience scientists had with chloroquine and its derivatives, it's quite unlikely for amodiaquine to proceed in human trials.

For the rest of the predicted interactions, we notice either a stationarity or a slight increase in the evidence score of each triplet.

A notably interesting prediction of the model is the slight increase of the score of the predicted triplet <imatinib, DOWNREGULATE, COVID-19/SARS-CoV-2>. The half-expert gave the corresponding pair a 2 as an evidence score 2 (pair that may or may not be true but is *not* worth investigating) in May 2020 and a 3 (pair that may or may not be true but is worth investigating) in September 2021. Indeed, during the last months, World Health Organisation (WHO) started a large scale clinical trial for imatinib as a possible treatment for SARS-CoV-2. The trial has recently reached phase 3 and the outcomes in terms of reduction of morbidity and mortality are very promising. [2]

[2] https://www.who.int/news/item/11-08-2021-who-s-solidarity-clinical-trial-enters-a-new-phase-with-three-new-candidate-drugs

5 Related Work

In this chapter we briefly present other types of knowledge embedding methods and also give more examples of link prediction applied to the fields of biology and medicine.

5.1 Other interesting knowledge embedding methods

5.1.1 Models based on random walks

An interesting category of knowledge embedding models are the ones that use random walks on the knowledge graph to learn their embeddings. Some of the most important work in this category includes *DeepWalk* [PAS14] and *Node2Vec* [GL16].

DeepWalk learns continuous low-dimensional node embeddings by using *local* information from truncated random walks and treating the walks as a graph equivalent of sentences by using Word2Vec (the Skip-gram model) [Mik+13a] on the sequence of nodes obtained by the random walk. An advantage of *DeepWalk* is its adaptability: it doesn't need to repeat the whole learning process to update its node embeddings as the network evolves over time.

Node2Vec also uses random walks to learn its continuous low-dimensional representations but it focuses a lot on the concept of the *neighbourhood* of nodes. It uses a biased random walk to explore neighbourhoods of nodes both in a BFS and in a DFS pattern, thus creating new, more diverse kinds of neighbourhoods. It then learns vector representations for the nodes that maximize the likelihood of preserving existing neighbourhoods.

5.1.2 Discrete embeddings

Another noteworthy category of knowledge embedding models are the ones that use discrete embeddings for their vector representations of entities and relations. They aim to achieve comparable results to the state-of-the-art continuous embedding methods but keeping their computational cost lower (especially regarding storage size of a model's

parameters). Interesting work in this category includes *B-CP* [Kis+19], *DKGE* [Li+21] and *TS-NL* [Sac20].

B-CP uses binary embeddings by introducing a quantization function into the optimization problem. The optimization is carried out in two steps. Firstly, the problem is relaxed by removing the discreteness constrains and continuous optimization is carried out. Then the quantization function is used to replace floating-point valued parameters with binarized ones. This approach however results in high quantization loss [Zha+16]. *DKGE* counters this problem by introducing a quantization function directly into the optimization problem they being solved during training. This approach gives better efficiency and accuracy, even compared to continuous embedding models, on dense datasets.

TS-NL is not a model on its own, but a method to compress continuous embeddings into discrete ones. It achieves $50-1000\times$ compression ratio with a minor performance loss. The models it was tested on were *TransE* [Bor+13], *DistMult* [Yan+15], *HolE* [NRP16], *ComplEX* [Tru+16], *ConvE* [MSR18], *RotatE* [Sun+19], *HypER* [BAH19b] and *TuckER* [BAH19a].

5.1.3 Using text descriptors of nodes

An interesting idea in knowledge embeddings is leveraging text descriptors of the nodes in addition to the structure of the graph. Pioneering work in this direction include *Content-Aware Node2Vec* [Kot20] and *KEPLER* [Wan+21].

Content-Aware Node2Vec, similarly to *Node2Vec* exploits wide network contexts by generating random walks to construct the network neighborhood of each node and then using the Skip-gram model to learn node embeddings that successfully predict the nodes in each walk. However, it leverages the textual information of each node by using a neural sequence encoder that produces each node's embedding $f(v)$ from the textual descriptor $S(v)$ of the node v.

KEPLER is a model for **K**nowledge **E**mbedding and **P**re-trained **L**anguag**E** **R**epresentation. In *KEPLER* textual entity descriptors are firstly encoded with a pre-trained language representation model and the resulting embeddings are then optimised jointly for Natural Language Processing (NLP) and knowledge embedding tasks. As a result, the model is not only capable of link prediction in knowledge graphs but it can also perform various NLP tasks as well, such as Relation Classification and Entity Typing. It also performs well on the General Language Understanding Evaluation (GLUE) benchmark [Wan+a].

5.2 Examples of knowledge graph embeddings in biology and medicine

An interesting application of knowledge graph embeddings is in using them for making safe medicine recommendations. The Safe Medicine Recommendation (SMR) framework [Gon+21] combines knowledge embeddings with Electronic Medical Records (EMRs) of patients to help doctors make better clinical decisions for their patients. SMR firstly constructs a high-quality heterogeneous graph by bridging EMRs and medical knowledge graphs together and then embeds this graph's entities (diseases, medicines, patients) and their corresponding relations. Finally, SMR uses the embeddings for link predicition representing the task of medicine recommendation to a certain patient.

Further research on using EMRs and biomedical knowledge graphs for medicine recommendations includes the *DARLING* (**D**emographic **A**ware p**R**obabi**L**istic med**I**cal k**N**owledge embeddin**G**) model [GKM21] which incorporates demographics in the medical entities to make more accurate drug recommendations.

Another noteworthy use of knowledge embeddings in the biomedical sector and more specifically in the fight against the COVID-19 pandemic is *TeX-Graph* [KS20] where embeddings are used in drug repurposing for COVID-19. This approach can help reduce the cost and time needed for exploring new possible cures for a novel disease.

6 Conclusions and Future work

In this chapter we conclude and provide ideas for future work.

6.1 Summary of the work of this thesis

In this thesis we firstly reconstructed a different version of the COVID-19 related knowledge graph created by Sotiris Kotitsas in his MSc thesis, while also making sure that we describe precisely each step of the construction of the graph to make our experiments reproducible. Afterwards, we experimented with several link prediction methods on benchmark datasets and on the aforementioned COVID-19 related graph and we chose one of the models (namely *TuckER*) to make predictions of interactions between SARS-CoV-2/COVID-19 and other biomedical entities. During the models' benchmarking proccess we also took into account the computational efficiency of each method, which is often overlooked in similar cases. Finally, we presented the top predictions of the model to a biomedical half-expert to evaluate whether link prediction can be used as a tool to help experts discover new knowledge.

6.2 Key takeaways

From the evaluation of our predictions by the half-expert we conclude that knowledge embeddings should most probably be used only to guide research towards certain directions and certainly not be used to make concrete predictions that may be used without further research. They should also be used with caution even when used to guide the experts' research since noisy data (or wrong predictions due to the model's poor modelling capacity) may lure experts in wrong or non-promising research.

6.3 Ideas for Future Work

Future extensions of this thesis could include showing our predictions to actual experts (biomedical scientists) to evaluate them. We could also ask them to evaluate the type and the direction of the relationships predicted.

It would also be interesting to check the approach of using link prediction methods to suggest new valuable directions of research for other more thoroughly studied and dangerous diseases as well (e.g. cancer, AIDS). This way we won't be training the models with noisy and incomplete data, but with manually curated data collected throughout a time period of many years/decades.

7 Appendix

7.1 Matchings of CHEMBL MoA to our four relationship types

Action Type	Mapped to
INHIBITOR	DOWNREGULATE
ANTAGONIST	DOWNREGULATE
AGONIST	UPREGULATE
BINDING AGENT	INTERACTS-WITH
BLOCKER	DOWNREGULATE
MODULATOR	UPREGULATE
POSITIVE ALLOSTERIC MODULATOR	UPREGULATE
HYDROLYTIC ENZYME	INTERACTS-WITH
PARTIAL AGONIST	UPREGULATE
ACTIVATOR	UPREGULATE
OPENER	UPREGULATE
OTHER	CAUSES-REACTION
CROSS-LINKING AGENT	INTERACTS-WITH
POSITIVE MODULATOR	UPREGULATE
SEQUESTERING AGENT	DOWNREGULATE
CHELATING AGENT	DOWNREGULATE
NEGATIVE ALLOSTERIC MODULATOR	DOWNREGULATE
INVERSE AGONIST	DOWNREGULATE
RELEASING AGENT	UPREGULATE
STABILISER	CAUSES-REACTION
ANTISENSE INHIBITOR	DOWNREGULATE
SUBSTRATE	CAUSES-REACTION
PROTEOLYTIC ENZYME	INTERACTS-WITH
DISRUPTING AGENT	DOWNREGULATE
RNAI INHIBITOR	DOWNREGULATE
OXIDATIVE ENZYME	INTERACTS-WITH
ALLOSTERIC ANTAGONIST	DOWNREGULATE
DEGRADER	DOWNREGULATE
REDUCING AGENT	INTERACTS-WITH

Tab. 7.1: CHEMBL 'mechanisms of action' and their mapping to the relationship types of our initial Coronaviradae graph.

7.2 Categories we restrict our predictions on

The list of categories we restrict our predictions on, is the following (quotes are used since some categories contain commas):

'Carbohydrate Sequence', 'Genetic Function', 'Body Space or Junction', 'Nucleic Acid, Nucleoside, or Nucleotide', 'Organophosphorus Compound', 'Tissue', 'Physiologic Function', 'Molecular Function', 'Clinical Drug', 'Cell or Molecular Dysfunction', 'Injury or Poisoning', 'Gene or Genome', 'Fully Formed Anatomical Structure', 'Element, Ion, or Isotope', 'Nucleotide Sequence', 'Laboratory Procedure', 'Body Substance', 'Cell Component', 'Chemical', 'Embryonic Structure', 'Body Part, Organ, or Organ Component', 'Hazardous or Poisonous Substance', 'Anatomical Abnormality', 'Eicosanoid', 'Hormone', 'Pathologic Function', 'Molecular Sequence', 'Cell', 'Body Location or Region', 'Inorganic Chemical', 'Molecular Biology Research Technique', 'Enzyme', 'Amino Acid, Peptide, or Protein', 'Cell Function', 'Amino Acid Sequence', 'Organic Chemical', 'Laboratory or Test Result', 'Body System', 'Neuroreactive Substance or Biogenic Amine', 'Lipid', 'Disease or Syndrome', 'Therapeutic or Preventive Procedure', 'Antibiotic', 'Biomedical or Dental Material', 'Carbohydrate', 'Steroid', 'Pharmacologic Substance', 'Immunologic Factor', 'Chemical Viewed Structurally', 'Organ or Tissue Function', 'Diagnostic Procedure', 'Vitamin', 'Anatomical Structure', 'Receptor', 'Chemical Viewed Functionally', 'Mental or Behavioral Dysfunction', 'Biologically Active Substance', 'Congenital Abnormality', 'Sign or Symptom', 'Indicator, Reagent, or Diagnostic Aid', 'Acquired Abnormality', 'Finding', 'Experimental Model of Disease', 'Neoplastic Process'

7.3 Final predictions

In the following table we mention the 91 final predictions with a probability score ≥ 0.05. For each prediction we also mention the probability score given to the triplet by our model.

Source node	Relationship type	Target node	Probability
SARS-CoV-2/COVID-19	UPREGULATE	rituximab	0.1349
SARS-CoV-2/COVID-19	UPREGULATE	dna-binding proteins	0.1272
SARS-CoV-2/COVID-19	UPREGULATE	tumor suppressor proteins	0.127
SARS-CoV-2/COVID-19	UPREGULATE	gaba-a receptor; anion channel	0.1165
SARS-CoV-2/COVID-19	UPREGULATE	injection product	0.1129
SARS-CoV-2/COVID-19	UPREGULATE	glycosyltransferase	0.0578
SARS-CoV-2/COVID-19	UPREGULATE	phosphate measurement	0.0552
fosamprenavir	DOWNREGULATE	SARS-CoV-2/COVID-19	0.5418
tamoxifen	DOWNREGULATE	SARS-CoV-2/COVID-19	0.429
ozenoxacin	DOWNREGULATE	SARS-CoV-2/COVID-19	0.4242
chloroquine hydrochloride	DOWNREGULATE	SARS-CoV-2/COVID-19	0.3306

Source node	Relationship type	Target node	Probability
telaprevir	DOWNREGULATE	SARS-CoV-2/COVID-19	0.3232
chloroquine sulfate	DOWNREGULATE	SARS-CoV-2/COVID-19	0.3196
etirinotecan pegol	DOWNREGULATE	SARS-CoV-2/COVID-19	0.2978
zanamivir	DOWNREGULATE	SARS-CoV-2/COVID-19	0.2932
imatinib	DOWNREGULATE	SARS-CoV-2/COVID-19	0.2917
cyclosporine	DOWNREGULATE	SARS-CoV-2/COVID-19	0.2795
amodiaquine	DOWNREGULATE	SARS-CoV-2/COVID-19	0.2747
mefloquine	DOWNREGULATE	SARS-CoV-2/COVID-19	0.2686
aminoquinoline anti-malarial (product)	DOWNREGULATE	SARS-CoV-2/COVID-19	0.2384
pegfilgrastim	DOWNREGULATE	SARS-CoV-2/COVID-19	0.2159
lovastatin	DOWNREGULATE	SARS-CoV-2/COVID-19	0.2145
eflornithine	DOWNREGULATE	SARS-CoV-2/COVID-19	0.2031
indinavir	DOWNREGULATE	SARS-CoV-2/COVID-19	0.1909
telotristat ethyl	DOWNREGULATE	SARS-CoV-2/COVID-19	0.1848
interferon omega 1	DOWNREGULATE	SARS-CoV-2/COVID-19	0.1845
temsirolimus	DOWNREGULATE	SARS-CoV-2/COVID-19	0.1772
rabbit anti-human t-lymphocyte globulin	DOWNREGULATE	SARS-CoV-2/COVID-19	0.1722
boceprevir	DOWNREGULATE	SARS-CoV-2/COVID-19	0.1709
oseltamivir phosphate	DOWNREGULATE	SARS-CoV-2/COVID-19	0.1677
galeterone	DOWNREGULATE	SARS-CoV-2/COVID-19	0.1642
primaquine phosphate	DOWNREGULATE	SARS-CoV-2/COVID-19	0.1376
suramin	DOWNREGULATE	SARS-CoV-2/COVID-19	0.1364
roxadustat	DOWNREGULATE	SARS-CoV-2/COVID-19	0.1331
halofantrine	DOWNREGULATE	SARS-CoV-2/COVID-19	0.1227
mebendazole	DOWNREGULATE	SARS-CoV-2/COVID-19	0.122
morphine sulfate	DOWNREGULATE	SARS-CoV-2/COVID-19	0.1218
elvitegravir	DOWNREGULATE	SARS-CoV-2/COVID-19	0.1154
acyclovir	DOWNREGULATE	SARS-CoV-2/COVID-19	0.1096
desflurane	DOWNREGULATE	SARS-CoV-2/COVID-19	0.1083
rebetron	DOWNREGULATE	SARS-CoV-2/COVID-19	0.1081
acetarsone	DOWNREGULATE	SARS-CoV-2/COVID-19	0.1053
alprostadil	DOWNREGULATE	SARS-CoV-2/COVID-19	0.1012
fenofibrate	DOWNREGULATE	SARS-CoV-2/COVID-19	0.0967
kinase inhibitor [epc]	DOWNREGULATE	SARS-CoV-2/COVID-19	0.0965
foscarnet	DOWNREGULATE	SARS-CoV-2/COVID-19	0.0961
ranibizumab	DOWNREGULATE	SARS-CoV-2/COVID-19	0.0955
upamostat	DOWNREGULATE	SARS-CoV-2/COVID-19	0.0937

Source node	Relationship type	Target node	Probability
varicella-zoster virus vaccine live (oka-merck) strain	DOWNREGULATE	SARS-CoV-2/COVID-19	0.0933
acetophenazine	DOWNREGULATE	SARS-CoV-2/COVID-19	0.0928
terconazole	DOWNREGULATE	SARS-CoV-2/COVID-19	0.0927
cpg-odn	DOWNREGULATE	SARS-CoV-2/COVID-19	0.0916
saracatinib	DOWNREGULATE	SARS-CoV-2/COVID-19	0.0875
protease inhibitors	DOWNREGULATE	SARS-CoV-2/COVID-19	0.0863
maribavir	DOWNREGULATE	SARS-CoV-2/COVID-19	0.0861
beta-thujaplicin	DOWNREGULATE	SARS-CoV-2/COVID-19	0.0857
recombinant interferon beta	DOWNREGULATE	SARS-CoV-2/COVID-19	0.0815
nafamostat	DOWNREGULATE	SARS-CoV-2/COVID-19	0.0811
acyclovir sodium	DOWNREGULATE	SARS-CoV-2/COVID-19	0.0809
varespladib methyl	DOWNREGULATE	SARS-CoV-2/COVID-19	0.0801
tecovirimat	DOWNREGULATE	SARS-CoV-2/COVID-19	0.0791
guanidine hydrochloride	DOWNREGULATE	SARS-CoV-2/COVID-19	0.0788
indinavir sulfate	DOWNREGULATE	SARS-CoV-2/COVID-19	0.0764
interferon gamma-1b	DOWNREGULATE	SARS-CoV-2/COVID-19	0.0757
nelfinavir mesylate	DOWNREGULATE	SARS-CoV-2/COVID-19	0.0746
teriflunomide	DOWNREGULATE	SARS-CoV-2/COVID-19	0.0734
aprotinin	DOWNREGULATE	SARS-CoV-2/COVID-19	0.0719
interferon alfa-2b / ribavirin	DOWNREGULATE	SARS-CoV-2/COVID-19	0.0697
simeprevir	DOWNREGULATE	SARS-CoV-2/COVID-19	0.0693
u 0126	DOWNREGULATE	SARS-CoV-2/COVID-19	0.0683
decanoyl rvkr chloromethylketone	DOWNREGULATE	SARS-CoV-2/COVID-19	0.0679
ticagrelor	DOWNREGULATE	SARS-CoV-2/COVID-19	0.0672
gemcitabine	DOWNREGULATE	SARS-CoV-2/COVID-19	0.065
vandetanib	DOWNREGULATE	SARS-CoV-2/COVID-19	0.0624
e 64	DOWNREGULATE	SARS-CoV-2/COVID-19	0.0616
telotristat etiprate	DOWNREGULATE	SARS-CoV-2/COVID-19	0.0605
rintatolimod	DOWNREGULATE	SARS-CoV-2/COVID-19	0.0593
brivudine	DOWNREGULATE	SARS-CoV-2/COVID-19	0.0592
linoleic acid	DOWNREGULATE	SARS-CoV-2/COVID-19	0.0586
doxycycline	DOWNREGULATE	SARS-CoV-2/COVID-19	0.0578
il1a protein, human	DOWNREGULATE	SARS-CoV-2/COVID-19	0.0564
thiothixene	DOWNREGULATE	SARS-CoV-2/COVID-19	0.0551

Source node	Relationship type	Target node	Probability
helichrysetin	DOWNREGULATE	SARS-CoV-2/COVID-19	0.0536
human c1-esterase inhibitor	DOWNREGULATE	SARS-CoV-2/COVID-19	0.0528
rotavirus vaccines	DOWNREGULATE	SARS-CoV-2/COVID-19	0.0527
ssya10-001	DOWNREGULATE	SARS-CoV-2/COVID-19	0.0526
proteasome inhibitor	DOWNREGULATE	SARS-CoV-2/COVID-19	0.0524
tamoxifen	UNIDIRECTIONAL	SARS-CoV-2/COVID-19	0.0976
acyclovir	UNIDIRECTIONAL	SARS-CoV-2/COVID-19	0.0737
glycosyltransferase	UNIDIRECTIONAL	SARS-CoV-2/COVID-19	0.0561
high occupancy target region	UNIDIRECTIONAL	SARS-CoV-2/COVID-19	0.0546

Tab. 7.2: Final predictions of interactions between SARS-CoV-2/COVID-19 and other biomedical entities.

7.4 Full results of half-expert evaluation

In the following table we list the 88 pairs we gave the half-expert to evaluate and his answers for each pair. Since in each pair the one entity was always SARS-CoV-2/COVID-19, we use only one column to define each pair.

Entity 2	Evidence score May 2020	Evidence score September 2021
rituximab	3	5
dna-binding proteins	GENERIC	GENERIC
tumor suppressor proteins	GENERIC	GENERIC
gaba-a receptor; anion channel	2	3
injection product	GENERIC	GENERIC
glycosyltransferase	2	2
phosphate measurement	2	3
fosamprenavir	2	3
tamoxifen	3	4
ozenoxacin	2	2
chloroquine hydrochloride	3	0
telaprevir	2	3
chloroquine sulfate	3	0
etirinotecan pegol	2	2
zanamivir	2	2
imatinib	3	4

Entity 2	Evidence score May 2020	Evidence score September 2021
cyclosporine	3	3
amodiaquine	3	1
mefloquine	2	3
aminoquinoline antimalarial (product)	GENERIC	GENERIC
pegfilgrastim	2	2
lovastatin	3	3
eflornithine	2	2
indinavir	2	3
telotristat ethyl	2	3
interferon omega 1	2	2
temsirolimus	2	3
rabbit anti-human t-lymphocyte globulin	2	3
boceprevir	2	3
oseltamivir phosphate	3	3
galeterone	2	2
primaquine phosphate	2	2
suramin	2	3
roxadustat	2	3
halofantrine	2	3
mebendazole	3	3
morphine sulfate	5	5
elvitegravir	2	3
acyclovir	2	3
desflurane	3	3
rebetron	2	2
acetarsone	2	2
alprostadil	2	3
fenofibrate	2	3
kinase inhibitor [epc]	GENERIC	GENERIC
foscarnet	2	2
ranibizumab	2	2
upamostat	2	3
varicella-zoster virus vaccine live (oka-merck) strain	2	2
acetophenazine	2	3
terconazole	2	2
cpg-odn	3	3
saracatinib	3	3

Entity 2	Evidence score May 2020	Evidence score September 2021
protease inhibitors	GENERIC	GENERIC
maribavir	2	2
beta-thujaplicin	2	3
recombinant interferon beta	2	3
nafamostat	3	3
acyclovir sodium	2	3
varespladib methyl	2	3
tecovirimat	2	2
guanidine hydrochloride	2	2
indinavir sulfate	2	3
interferon gamma-1b	2	3
nelfinavir mesylate	3	3
teriflunomide	2	3
aprotinin	2	3
interferon alfa-2b/ribavirin	3	3
simeprevir	3	3
u 0126	2	2
decanoyl rvkr chloromethylketone	2	3
ticagrelor	3	3
gemcitabine	3	3
vandetanib	2	2
e 64	3	3
telotristat etiprate	2	2
rintatolimod	3	3
brivudine	2	3
linoleic acid	2	3
doxycycline	3	3
il1a protein, human	2	3
thiothixene	2	2
helichrysetin	2	3
human c1-esterase inhibitor	2	3
rotavirus vaccines	NOISE	NOISE
ssya10-001	3	3
proteasome inhibitor	3	3
high occupancy target region	NOISE	NOISE

Tab. 7.3: Evaluation of pairs by the biomedical half-expert. Since Entity 1 is always SARS-CoV-2/COVID-19, we only need Entity 2 to define a pair.

Bibliography

[AA01] Lada Adamic and Eytan Adar. "Friends and Neighbors on the Web". In: *Social Networks* 25 (2001), pp. 211–230.

[BAH19a] I. Balazevic, C. Allen, and T. Hospedales. "TuckER: Tensor Factorization for Knowledge Graph Completion". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (2019).

[BAH19b] Ivana Balažević, Carl Allen, and Timothy M Hospedales. "Hypernetwork knowledge graph embeddings". In: *International Conference on Artificial Neural Networks* (2019).

[Bor+11] Antoine Bordes, Jason Weston, Ronan Collobert, and Yoshua Bengio. "Learning Structured Embeddings of Knowledge Bases". In: *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence* (2011).

[Bor+13] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Durán, Jason Weston, and Oksana Yakhnenko. "Translating embeddings for modelling multi-relational data". In: *Proceedings of the 26th International Conference on Neural Information Processing Systems* (2013).

[Det+18] T. Dettmers, P. Minervini, P. Stenetorp, and S. Riedel. "Convolutional 2D Knowledge Graph Embeddings". In: *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*. 2018, pp. 1811–1818.

[Fel98] C. Fellbaum, ed. *WordNet: An Electronic Lexical Database*. MIT Press, 1998.

[Fri+99] Nir Friedman, Lise Getoor, Daphne Koller, and Avi Pfeffer. "Learning Probabilistic Relational Models". In: *International Joint Conference on Artificial Intelligence* (1999).

[GKM21] Aynur Guluzade, Endri Kacupaj, and Maria Maleshkova. "Demographic Aware Probabilistic Medical Knowledge Graph Embeddings of Electronic Medical Records". In: *Artificial Intelligence in Medicine 2021 (AIME 2021)* (2021).

[GL16] Aditya Grover and Jure Leskovec. "node2vec: Scalable Feature Learning for Networks". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2016).

[Glo+13] Xavier Glorot, Antoine Bordes, Jason Weston, and Yoshua Bengio. "A Semantic Matching Energy Function for Learning with Multi-relational Data". In: *Machine Learning* (2013).

[Gon+21] Fan Gong, Meng Wang, Haofen Wang, Sen Wang, and Mengyue Liu. "SMR: Medical Knowledge Graph Embedding for Safe Medicine Recommendation". In: *Big Data Research, Volume 23* (2021).

[Ji+15] Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. "Knowledge Graph Embedding via Dynamic Mapping Matrix". In: *Association for Computational Linguistics* (2015).

[Kar+19] Md. Rezaul Karim, Michael Cochez, Joao Bosco Jares, Mamtaz Uddin, and Oya Beyan Stefan Decker. "Drug-Drug Interaction Prediction Based on Knowledge Graph Embeddings and Convolutional-LSTM Network". In: *Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics* (2019).

[Kis+19] Koki Kishimoto, Katsuhiko Hayashi, Genki Akai, Masashi Shimbo, and Kazunori Komatani. "Binarized Knowledge Graph Embeddings". In: *ECIR 2019: Advances in Information Retrieval* (2019).

[Kot20] Sotiris Kotitsas. "Neural Graph Representations and their Application to Link Prediction". In: *Athens University of Economics and Business, Department of Informatics, Master in Computer Science* (2020).

[KS20] Charilaos I. Kanatsoulis and Nicholas D. Sidiropoulos. "TeX-Graph: Coupled tensor-matrix knowledge-graph embedding for COVID-19 drug repurposing". In: *Proceedings of the 2021 Society of Industrial and Applied Mathematics International Conference on Data Mining (SDM)* (2020).

[Li+21] Yunqi Li, Shuyuan Xu, Bo Liu, Zuohui Fu, Shuchang Liu, Xu Chen, and Yongfeng Zhang. "Discrete Knowledge Graph Embedding based on Discrete Optimization". In: *AAAI-20 Workshop on Knowledge Discovery from Unstructured Data in Financial Services* (2021).

[Lin+15] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. "Learning Entity and Relation Embeddings for Knowledge Graph Completion". In: *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence* (2015).

[LZ11] Linyuan Lü and Tao Zhou. "Link prediction in complex networks: A survey". In: *Physica A 390* (2011).

[Mik+13a] Tomas Mikolov, G.s. Corrado, Kai Chen, and Jeffrey Dean. "Efficient Estimation of Word Representations in Vector Space". In: *Proceedings of the International Conference on Learning Representations* (2013).

[Mik+13b] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. "Distributed representations of words and phrases and their compositionality". In: *Advances in neural information processing systems* (2013).

[MSR18] Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. “Convolutional 2D Knowledge Graph Embeddings”. In: *AAAI* (2018).

[Ngu+18] Dai Q. Nguyen, T. Nguyen, Dat Q. Nguyen, and D. Q. Phung. “A Novel Embedding Model for Knowledge Base Completion Based on Convolutional Neural Network”. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)* (2018), pp. 327–333.

[NRP16] Maximilian Nickel, Lorenzo Rosasco, and Tomaso Poggio. “Holographic Embeddings of Knowledge Graphs”. In: *AAAI* (2016).

[PAS14] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. “DeepWalk: Online Learning of Social Representations”. In: *KDD ’14: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining* (2014).

[Sac20] Mrinmaya Sachan. “Knowledge Graph Embedding Compression”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (2020).

[Sun+19] Z. Sun, Z.-H. Deng, J.-Y. Nie, and J. Tang. “RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space”. In: *International Conference on Learning Representations*. 2019.

[Tan+20] Yun Tang, Jing Huang, Guangtao Wang, Xiadong He, and Bowen Zhou. “Orthogonal Relation Transforms with Graph Context Modeling for Knowledge Graph Embedding”. In: 2020.

[TC15] Kristina Toutanova and Danqi Chen. “Observed versus latent features for knowledge base and text inference”. In: *ACL* (2015).

[Tru+16] Théo Truillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. “Complex Embeddings for Simple Link Prediction”. In: *International Conference on Machine Learning* (2016).

[Tuc64] LR Tucker. “The Extension of Factor Analysis to Three-Dimensional Matrices”. In: *Contributions to Mathematical Psychology* (1964).

[Wan+a] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. “GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding”. In: *ICLR 2019* ().

[Wan+b] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. “Knowledge graph embedding: A survey of approaches and applications”. In: *TKDE, 29:2724–2743* ().

[Wan+14] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. “Knowledge Graph Embedding by Translating on Hyperplanes”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 28 (2014).

[Wan+21] Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. "KEPLER: A Unified Model for Knowledge Embedding and Pre-trained Language Representation". In: *Transactions of the Association for Computational Linguistics* (2021).

[WZ] Yuhao Wang and Jianyang Zeng. "Predicting drug-target interactions using restricted Boltzmann machines". In: *Bioinformatics* 29 (), pp. 126–134.

[Yan+15] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. "Embedding entities and relations for learning and inference in knowledge bases". In: *ICLR* (2015).

[Yan+20] Fang Yang, Kunjie Fan, Dandan Song, and Huakang Lin. "Graph-based prediction of Protein-protein interactions with attributed signed graph embedding". In: *BMC Bioinformatics* (2020).

[Yas+20] Hajime Yasuda, Yutaka Tsukune, Naoki Watanabe, Kazuya Sugimoto, Ayana Uchimura, Misa Tateyama, Yosuke Miyashita, Yusuke Ochi, and Norio Komatsu. "Persistent COVID-19 Pneumonia and Failure to Develop Anti-SARS-CoV-2 Antibodies During Rituximab Maintenance Therapy for Follicular Lymphoma". In: *Clin Lymphoma Myeloma Leuk* (2020).

[ZAL18] Marika Zitnik, Monica Agrawal, and Jure Leskovec. "Modeling polypharmacy side effects with graph convolutional networks". In: *Bioinformatics* 34 (2018), pp. 457–466.

[Zha+16] Hanwang Zhang, Fumin Shen, Wei Liu, Xiangnan He, Huanbo Luan, and Tat-Seng Chua. "Discrete Collaborative Filtering". In: *SIGIR '16: Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval* (2016).

[Zha+19] Shuai Zhang, Yi Tay, Lina Yao, and Qi Liu. "Quaternion Knowledge Graph Embeddings". In: *NeurIPS* (2019).