Retrofitting Distributional Embeddings to Knowledge Graphs with Functional Relations

Benjamin J. Lengerich
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
blengeri@cs.cmu.edu

Andrew L. Maas and **Christopher Potts**
Stanford University, Roam Analytics
195 East 4th Avenue
San Mateo, CA 94401
{amaas, cgpotts}@roaminsight.com

Abstract

Knowledge graphs are a versatile framework to encode richly structured data relationships, but it can be challenging to combine these graphs with unstructured data. Methods for retrofitting pre-trained entity representations to the structure of a knowledge graph typically assume that entities are embedded in a connected space and that relations imply similarity. However, useful knowledge graphs often contain diverse entities and relations (with potentially disjoint underlying corpora) which do not accord with these assumptions. To overcome these limitations, we present *Functional Retrofitting*, a framework that generalizes current retrofitting methods by explicitly modeling pairwise relations. Our framework can directly incorporate a variety of pairwise penalty functions previously developed for knowledge graph completion. Further, it allows users to encode, learn, and extract information about relation semantics. We present both linear and neural instantiations of the framework. Functional Retrofitting significantly outperforms existing retrofitting methods on complex knowledge graphs and loses no accuracy on simpler graphs (in which relations do imply similarity). Finally, we demonstrate the utility of the framework by predicting new drug–disease treatment pairs in a large, complex health knowledge graph.

1 Introduction

Distributional representations of concepts are often easy to obtain from unstructured data sets, but they tend to provide only a blurry picture of the relationships that exist between concepts. In contrast, knowledge graphs directly encode this relational information, but it can be difficult to summarize the graph structure in a single representation for each entity.

To combine the advantages of distributional and relational data, Faruqui et al. (2015) propose to *retrofit* embeddings learned from distributional data to the structure of a knowledge graph. Their method first learns entity representations based solely on distributional data and then applies a retrofitting step to update the representations based on the structure of a knowledge graph. This modular approach conveniently separates the distributional data and entity representation learning from the knowledge graph and retrofitting model, allowing one to flexibly combine, reuse, and adapt existing representations to new tasks.

However, a core assumption of Faruqui et al. (2015)'s retrofitting model is that connected entities should have similar embeddings. This assumption often fails to hold in large, complex knowledge graphs, for a variety of reasons. First, subgraphs of a knowledge graph often contain distinct classes of entities that are most naturally embedded in disconnected vector spaces. In the extreme case, the representations for these entities might derive from very different underlying data sets. For example, in a health knowledge graph, the subgraphs containing diseases and drugs should be allowed to form disjoint vector spaces, and we might want to derive the initial representations from radically different data sets. Second, many knowledge graphs contain diverse relationships whose semantics are different from – perhaps even in conflict with – similarity. For instance, in the knowledge graph in Figure 1, the model

Proceedings of the 27th International Conference on Computational Linguistics, pages 2423–2436
Santa Fe, New Mexico, USA, August 20-26, 2018.

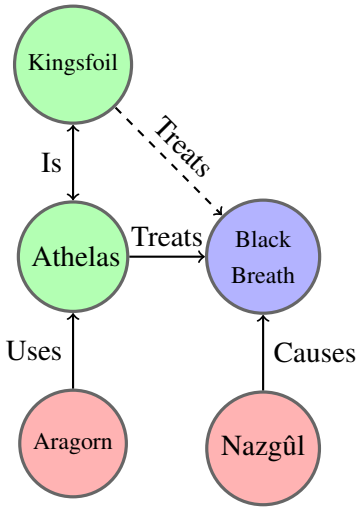

Figure 1: Toy knowledge graph with diverse relation types that connect treatments (green), diseases (blue), and persons (red) by known (solid) and unknown (dashed) relations. Traditional methods, which assume that all relations imply similarity, would retrofit Aragorn and Nazgûl toward similar embeddings.

of Faruqui et al. (2015) would model embeddings $\boldsymbol{q_{Aragorn}} \approx \boldsymbol{q_{Athelas}} \approx \boldsymbol{q_{BlackBreath}} \approx \boldsymbol{q_{Nazgûl}}$, which is problematic as Aragorn is not semantically similar to a Nazgûl (they are enemies).

To address these limitations, we present *Functional Retrofitting*, a retrofitting framework that explicitly models pairwise relations as functions. The framework supports a wide range of different instantiations, from simple linear relational functions to complex multilayer neural ones. Here, we evaluate both linear and neural instantiations of Functional Retrofitting on a variety of diverse knowledge graphs. For benchmarking against existing approaches, we use FrameNet and WordNet. We then move into the medical domain, where knowledge graphs play an important role in knowledge accumulation and discovery. These experiments show that even simple instantiations of Functional Retrofitting significantly outperform baselines on knowledge graphs with semantically complex relations and sacrifice no accuracy on graphs where Faruqui et al. (2015)'s assumptions about similarity do hold. Finally, we use the model to identify promising new disease targets for existing drugs.

Code which implements Functional Retrofitting is available at `https://github.com/roaminsight/roamresearch`.

2 Notation

A knowledge graph $\mathcal{G}$ is composed of a set of vertices $\mathcal{V}$, a set of relation types $\mathcal{R}$, and a set of edges $\mathcal{E}$ where each edge $e \in \mathcal{E}$ is a tuple (i, j, r) in which the relationship $r \in \mathcal{R}$ holds between vertices $i \in \mathcal{V}$ and $j \in \mathcal{V}$. Our goal is to learn a set of representations $\mathcal{Q} = \{\boldsymbol{q_i} : i \in \mathcal{V}\}$ which contain the information encoded in both the distributional data and the knowledge graph structure, and can be used for downstream analysis. Throughout this paper, we use a to refer to a scalar, $\boldsymbol{a}$ to refer to a vector, and $\boldsymbol{A}$ to refer to a matrix or tensor.

3 Related Work

Here we are interested in *post-hoc* retrofitting methods, which adjust entity embeddings to fit the structure of a previously unseen knowledge graph.

3.1 Retrofitting Models

The primary introduction of retrofitting was Faruqui et al. (2015), in which the authors showed the value of retrofitting semantic embeddings according to minimization of the weighted least squares problem

$$\Psi_{\mathcal{G}}(\mathcal{Q}) = \sum_{i \in \mathcal{V}} \alpha_i \left\| \boldsymbol{q_i} - \hat{\boldsymbol{q}}_{\boldsymbol{i}} \right\|^2 + \sum_{(i,j,r) \in \mathcal{E}} \beta_{ij} \left\| \boldsymbol{q_i} - \boldsymbol{q_j} \right\|^2 \tag{1}$$

where $\hat{\mathcal{Q}} = \{\hat{\boldsymbol{q}}_{\boldsymbol{i}} : i \in \mathcal{V}\}$ is the embedding learned from the distributional data and α_i, β_{ij} set the relative weighting of each type of data. When $\alpha_i = 1$ and $\beta_{ij} = \frac{1}{degree(i)}$, this model assigns equal weight to the distributional data and the structure of the knowledge graph.

More recently, Hamilton et al. (2017) presented GraphSAGE, a two-step method which learns both an aggregation function $f : \mathbb{R}^{d \times n} \rightarrow \mathbb{R}^k$, to condense the representations of neighbors into a single point, and an update function $g : \mathbb{R}^{k+d} \rightarrow \mathbb{R}^d$, to combine the aggregation with a central vertex. Here, d is the embedding dimensionality, k is the aggregation dimensionality, and $n > 0$ is the number of neighbors for each vertex. Note that $k > d$ is permitted, allowing for aggregation by concatenation. While this method is extremely effective for learning representations on simple knowledge graphs, it is not formulated for knowledge graphs with multiple types of relations. Furthermore, when the representation of a relation is known *a priori*, it can be useful to explicitly set the penalty function (e.g., Mrkšic et al. (2016) use hand-crafted functions to effectively model antonymy and synonymy). By aggregating neighbors into a point estimate before calculating relationship likelihoods, GraphSAGE makes it difficult to encode, learn, or extract the representation of a pairwise relation.

In a similar vein, Faruqui et al. (2016) developed a graph-based semi-supervised learning method to expand morpho-syntactic lexicons from seed sets. Though the task is different from the retrofitting task we consider here, the performance and scalability of their method demonstrate the utility of directly encoding pairwise relations as message-passing functions.

3.2 Relational Penalty Functions

Our new Functional Retrofitting framework models each relation via a penalty function $f_r : \mathbb{R}^{d_i+d_j} \rightarrow \mathbb{R}_{\geq 0}$ acting on a pair of entities (i, j) with embedding dimensionalities d_i and d_j, respectively. By explicitly modeling relations between pairs of entities, Functional Retrofitting supports the use of a wide array of scoring functions that have previously been developed for knowledge graph completion. Here, we present a brief review of such scoring functions; for an extensive review, see (Nickel et al., 2016).

TransE (Bordes et al., 2013) uses additive relations in which the penalty function $f_r(\boldsymbol{q_i}, \boldsymbol{q_j}) = \|\boldsymbol{q_i} + \boldsymbol{a_r} - \boldsymbol{q_j}\|_2^2$ is low iff $(i, j, r) \in \mathcal{E}$. The simple Unstructured Model (Bordes et al., 2012) was proposed as a naïve version of TransE that assigns all $\boldsymbol{a_r} = \mathbf{0}$, leading to the penalty function $f_r(\boldsymbol{q_i}, \boldsymbol{q_j}) = \|\boldsymbol{q_i} - \boldsymbol{q_j}\|_2^2$. This is the underlying penalty function of (Faruqui et al., 2015). It cannot consider multiple types of relations. In addition, while it models 1-to-1 relations well, it struggles to model multivalued relations.

TransH (Wang et al., 2014) was proposed to address this limitation by using multiple representations for a single entity via relation hyperplanes. For a relation r, TransH models the relation as a vector $\boldsymbol{a_r}$ on a hyperplane defined by normal vector $\boldsymbol{w_r}$. For a triple $(i, j, r) \in \mathcal{E}$, the entity embeddings $\boldsymbol{q_i}$ and $\boldsymbol{q_j}$ are first projected to the hyperplane of $\boldsymbol{w_r}$. By constraining $\|\boldsymbol{w_r}\|_2 = 1$, we have the penalty function $f_r(\boldsymbol{q_i}, \boldsymbol{q_j}) = \|g_r(\boldsymbol{q_i}) + \boldsymbol{a_r} - g_r(\boldsymbol{q_j})\|_2^2$ where $g_r(\boldsymbol{x}) = \boldsymbol{x} - \boldsymbol{w_r}^T \boldsymbol{x} \boldsymbol{w_r}$.

TransR (Lin et al., 2015) embeds relations in a separate space from entities by a relation-specific matrix $\boldsymbol{M_r} \in \mathbb{R}^{d \times k}$ that projects from entity space to relation space and a shared relation vector $\boldsymbol{a} \in \mathbb{R}^k$ that translates in relation space by $f_r(\boldsymbol{q_i}, \boldsymbol{q_j}) = \|\boldsymbol{q_i}\boldsymbol{M_r} + \boldsymbol{a} - \boldsymbol{q_j}\boldsymbol{M_r}\|_2^2$. We use this model as the inspiration for our linear penalty function.

The Neural Tensor Network (NTN; Socher et al. (2013)) defines a score function $f_r(\boldsymbol{q_i}, \boldsymbol{q_j}) = \boldsymbol{u_r}^T g(\boldsymbol{q_i}^T \boldsymbol{M_r} \boldsymbol{q_j} + \boldsymbol{M_{r,1}} \boldsymbol{q_i} + \boldsymbol{M_{r,2}} \boldsymbol{q_j} + \boldsymbol{b_r})$ where $\boldsymbol{u_r}$ is a relation-specific linear layer, $g : \mathbb{R}^k \rightarrow \mathbb{R}^k$ is the tanh operation applied element-wise, $\boldsymbol{M_r} \in \mathbb{R}^{d \times d \times k}$ is a 3-way tensor, and $\boldsymbol{M_{r,1}}, \boldsymbol{M_{r,2}} \in \mathbb{R}^{k \times d}$ are weight matrices. All of these models can be directly incorporated in our Functional Retrofitting framework.

4 Functional Retrofitting

We propose the framework of Functional Retrofitting (FR) to incorporate a set $\mathcal{F}$ of relation-specific penalty functions $f_r : \mathbb{R}^{d_i+d_j} \rightarrow \mathbb{R}_{\geq 0}$ which penalizes embeddings of entities i, j with dimensionalities

d_i, d_j, respectively. This gives the complete minimization:

$$\Psi_{\mathcal{G}}(\mathcal{Q};\mathcal{F}) = \sum_{i\in\mathcal{Q}} \alpha_i ||\boldsymbol{q_i} - \hat{\boldsymbol{q}}_{\boldsymbol{i}}||^2 + \sum_{(i,j,r)\in\mathcal{E}} \beta_{i,j,r} f_r(\boldsymbol{q_i}, \boldsymbol{q_j}) - \sum_{(i,j,r)\in\mathcal{E}^-} \beta_{i,j,r} f_r(\boldsymbol{q_i}, \boldsymbol{q_j}) + \sum_{r\in\mathcal{R}} \rho_\lambda(f_r) \tag{2}$$

where $\hat{\boldsymbol{q}}_{\boldsymbol{i}}$ is observed from distributional data, α_i and $\beta_{i,j,r}$ set the relative strengths of the distributional data and the knowledge graph structure, and ρ regularizes f_r with strength set by λ. $\mathcal{E}^-$ is the *negative space* of the knowledge graph, a set of edges that are not annotated in the knowledge graph. FR uses $\mathcal{E}^-$ to penalize relations that are implied by the representations but not annotated in the graph. To populate $\mathcal{E}^-$, we sample a single negative edge (i, j', r) with the same outgoing vertex for each true edge $(i, j, r) \in \mathcal{E}$. The user can calibrate trust in the completeness of the knowledge graph via the β hyperparameter.

In contrast to prior retrofitting work, FR explicitly encodes directed relations. This allows the model to fit graphs which contain diverse relation types and entities embedded in disconnected vector spaces. Here, we compare the performance of two instantiations of FR – one with all linear relations and one with all neural relations – and show that even these simple models provide significant performance improvements. In practice, we recommend that users select relation-specific functions in accordance with the semantics of their graph's relations.

4.1 Linear Relations

We implement a linear relational penalty function $f_r(\boldsymbol{q_i}, \boldsymbol{q_j}) = \|\boldsymbol{A_r q_j} + \boldsymbol{b_r} - \boldsymbol{q_i}\|^2$ with ℓ_2 regularization for minimization of:

$$\begin{aligned}\Psi_{\mathcal{G}}(\mathcal{Q};\mathcal{F}) = &\sum_{i=1}^{n} \alpha_i \left\|\boldsymbol{q_i} - \hat{\boldsymbol{q}}_{\boldsymbol{i}}\right\|^2 + \sum_{(i,j,r)\in\mathcal{E}} \beta_{i,j,r} \left\|\boldsymbol{A_r q_j} + \boldsymbol{b_r} - \boldsymbol{q_i}\right\|^2 \\ &- \sum_{(i,j,r)\in\mathcal{E}^-} \beta_{i,j,r} \left\|\boldsymbol{A_r q_j} + \boldsymbol{b_r} - \boldsymbol{q_i}\right\|^2 + \lambda \sum_{r\in\mathcal{R}} ||\boldsymbol{A_r}||^2 \end{aligned} \tag{3}$$

Identity Relations

Faruqui et al. (2015)'s model is a special case of this formulation in which

$$\boldsymbol{A_r} = \boldsymbol{I}, \quad \boldsymbol{b_r} = \boldsymbol{0} \quad \forall r, \qquad \beta_{i,j,r} = \begin{cases} \frac{1}{degree(i)} & (i,j,r) \in \mathcal{E} \\ 0 & (i,j,r) \in \mathcal{E}^- \end{cases}$$

Throughout the remainder of this paper, we refer to this baseline model as the "FR-Identity" retrofitting method.

Initialization

We initialize embeddings as those learned from distributional data and relations to imply similarity:

$$\boldsymbol{A_r} = \boldsymbol{I}, \quad \boldsymbol{b_r} = \boldsymbol{0} \quad , \quad \alpha_i = \begin{cases} 0 & \hat{\boldsymbol{q}}_{\boldsymbol{i}} = \boldsymbol{0} \\ \alpha & \hat{\boldsymbol{q}}_{\boldsymbol{i}} \neq \boldsymbol{0} \end{cases}, \quad \beta_{i,j,r} = \begin{cases} \frac{\beta^+}{d_r(i)} & (i,j,r) \in \mathcal{E} \\ \frac{\beta^-}{d_r(i)} & (i,j,r) \in \mathcal{E}^- \end{cases}$$

where $d_r(i)$ is the out-degree of vertex i for relation type r, α is a hyperparameter to trade off distributional data against structural data, and β sets the trust in completeness of the knowledge graph structure. In our experiments, we use $\beta^+ = 1,\ \beta^- = 0$ for straightforward comparison with the method of Faruqui et al. (2015) and optimize α by cross-validation. Given prior knowledge about the semantic meaning of relations, we could initialize relations to respect these meanings (e.g., antonymy could be represented by $\boldsymbol{A_r} = -\boldsymbol{I}$).

Learning Procedure

We optimize this model by block optimization. Conveniently, we have closed-form solutions where the partial derivatives of Eq. 3 equal 0:

$$\boldsymbol{b_r} = \frac{\sum_{(i,j)} (-1)^{I_{\{(i,j,r)\notin\mathcal{E}\}}} \beta_{i,j,r}(\boldsymbol{A_r q_j} - \boldsymbol{q_i})}{\sum_{(i,j)} (-1)^{I_{\{(i,j,r)\notin\mathcal{E}\}}} \beta_{i,j,r}} \tag{4}$$

$$\tilde{\boldsymbol{A}}_{\boldsymbol{r}} = \boldsymbol{U}\boldsymbol{V}^{-1} \tag{5}$$

$$\boldsymbol{U} = \sum_{(i,j):(i,j,r)\in\mathcal{E}} \beta_{i,j,r}(\boldsymbol{q_i} - \boldsymbol{b_r})\boldsymbol{q_j}^T - \sum_{(i,j):(i,j,r)\in\mathcal{E}^-} \beta_{i,j,r}(\boldsymbol{q_i} - \boldsymbol{b_r})\boldsymbol{q_j}^T \tag{6}$$

$$\boldsymbol{V} = \sum_{(i,j):(i,j,r)\in\mathcal{E}} \beta_{i,j,r}\boldsymbol{q_j}\boldsymbol{q_j}^T - \sum_{(i,j):(i,j,r)\in\mathcal{E}^-} \beta_{i,j,r}\boldsymbol{q_j}\boldsymbol{q_j}^T + \lambda\boldsymbol{I} \tag{7}$$

Constraining $\boldsymbol{A_r}$ to be orthogonal by $\boldsymbol{A_r} = \tilde{\boldsymbol{A}}_{\boldsymbol{r}}(\tilde{\boldsymbol{A}}_{\boldsymbol{r}}^T\tilde{\boldsymbol{A}}_{\boldsymbol{r}})^{-1/2}$, we have $\boldsymbol{q_i} = \frac{\boldsymbol{a_i}}{b_i}$ where

$$\begin{aligned}\boldsymbol{a_i} = \alpha_i\hat{\boldsymbol{q}}_{\boldsymbol{i}} &+ \sum_{(j,r):(i,j,r)\in\mathcal{E}} \beta_{i,j,r}(\boldsymbol{A_r q_j} + \boldsymbol{b_r}) + \sum_{(j,r):(j,i,r)\in\mathcal{E}} \beta_{j,i,r}\boldsymbol{A_r}^T(\boldsymbol{q_j} - \boldsymbol{b_r}) \\ &- \sum_{(j,r):(i,j,r)\in\mathcal{E}^-} \beta_{i,j,r}(\boldsymbol{A_r q_j} + \boldsymbol{b_r}) - \sum_{(j,r):(j,i,r)\in\mathcal{E}^-} \beta_{j,i,r}\boldsymbol{A_r}^T(\boldsymbol{q_j} - \boldsymbol{b_r})\end{aligned} \tag{8}$$

$$b_i = \alpha_i + \sum_{(j,r):(i,j,r)\in\mathcal{E}} \beta_{i,j,r} + \sum_{(j,r):(j,i,r)\in\mathcal{E}} \beta_{j,i,r} - \sum_{(j,r):(i,j,r)\in\mathcal{E}^-} \beta_{i,j,r} - \sum_{(j,r):(j,i,r)\in\mathcal{E}^-} \beta_{j,i,r} \tag{9}$$

4.2 Neural Relations

We also instantiate FR with a neural penalty function $f_r(\boldsymbol{q_i}, \boldsymbol{q_j}) = \sigma(\boldsymbol{q_i}^T\boldsymbol{A_r q_j})$ where σ is the element-wise tanh operation, $\boldsymbol{A_r} \in \mathbb{R}^{d_i \times d_j}$, again with ℓ_2 regularization.We initialize weights in a similar manner as for the linear relations and update via stochastic gradient descent. In our experiments, we use $\beta^+ = \beta^- = 1$, and sample the same number of non-neighbors as true neighbors.

5 Experiments

We test FR on four knowledge graphs. The first two are standard lexical knowledge graphs (FrameNet, WordNet) in which FR significantly improves retrofitting quality on complex graphs and loses no accuracy on simple graphs. The final two graphs are large healthcare ontologies (SNOMED-CT, Roam Health Knowledge Graph), which demonstrate the scalability of the framework and the utility of the new embeddings.

For each graph, we successively evaluate link prediction accuracy after retrofitting to links of other relation types. Specifically, for each relation type $r \in \mathcal{R}$, we retrofit to $\mathcal{G}_{\setminus r} = (\mathcal{V}, \mathcal{E}_{\setminus r})$ where $\mathcal{E}_{\setminus r} = \{(i,j,r') : (i,j,r') \in \mathcal{E}, r' \neq r\}$ is the set of edges with all relations of type r removed. After retrofitting, we train a Random Forest classifier to predict the presence of relation r between entities i and j (with 70% of vertices selected as training examples and the remainder reserved for testing). To have balanced class labels, we sample an equivalent number of non-edges, $\mathcal{E}_r^- = \{(i,j,r) : (i,j,r) \notin \mathcal{E}\}$ with $|\mathcal{E}_r^-| = |\mathcal{E}|$ and $|\{j : (i,j,r) \in \mathcal{E}_r^-\}| = |\{j : (i,j,r) \in \mathcal{E}\}| \; \forall \; i$. Thus, the random baseline classification rate is set to 50%. Other baselines are the embeddings built from distributional data and the retrofitting method of Faruqui et al. (2015), denoted as "None" and "FR-Identity", respectively.

5.1 FrameNet

FrameNet (Baker et al., 1998; Fillmore et al., 2003) is a linguistic knowledge graph containing information about lexical and predicate argument semantics of the English language. FrameNet contains two distinct entity classes: *frames* and *lexical units*, where a *frame* is a meaning and a *lexical unit* is a single meaning for a word. To create a graph from FrameNet, we connect lexical unit i to frame j if i occurs

in j. We denote this relation as "Frame", and its inverse "Lexical unit". Finally, we connect frames by the structure of FrameNet (Table 6). Distributional embeddings are from the Google News pre-trained Word2Vec model (Mikolov et al., 2013a); the counts of each entity type that were also found in the distributional corpus are shown in Table 6.

Results

As seen in Table 1, the representations learned by FR-Linear and FR-Neural are significantly more useful for link prediction than those of the baseline methods.

Retrofitting Model	**'Inheritance'** (2132/992)	**'Using'** (1552/668)	**'Reframing Mapping'** (544/312)	**'Subframe'** (356/168)	**'Perspective On'** (336/148)
None	87.58 ± 1.04	88.59 ± 1.93	85.60 ± 1.80	91.24 ± 0.86	89.59 ± 3.25
FR-Identity	90.79 ± 0.69	87.87 ± 1.48	87.02 ± 0.63	94.50 ± 1.70	94.24 ± 1.02
FR-Linear	$\mathbf{92.92 \pm 0.16}$	92.04 ± 1.45	89.37 ± 2.45	94.65 ± 1.05	$\mathbf{94.73 \pm 1.12}$
FR-Neural	92.46 ± 0.67	$\mathbf{92.54 \pm 1.45}$	$\mathbf{89.57 \pm 0.70}$	$\mathbf{95.65 \pm 2.21}$	94.04 ± 0.58

Retrofitting Model	**'Precedes'** (220/136)	**'See Also'** (268/76)	**'Causative Of'** (204/36)	**'Inchoative Of'** (60/16)
None	87.30 ± 4.33	85.11 ± 3.20	86.11 ± 6.00	82.50 ± 14.29
FR-Identity	85.26 ± 4.46	83.81 ± 2.14	84.49 ± 8.72	78.33 ± 20.14
FR-Linear	87.00 ± 2.18	91.93 ± 1.06	92.09 ± 6.34	82.50 ± 14.29
FR-Neural	$\mathbf{89.16 \pm 5.60}$	$\mathbf{93.25 \pm 1.79}$	$\mathbf{94.33 \pm 4.68}$	$\mathbf{85.00 \pm 7.07}$

Table 1: Retrofitting to FrameNet. Reported values are mean and standard deviation of the link prediction accuracies over three experiments. The number of edges used for (training/testing) is shown below each edge type.

5.2 WordNet

WordNet (Miller, 1995; Fellbaum, 2005) is a lexical database consisting of words (lemmas) which are grouped into unordered sets of synonyms (synsets). To examine the performance of FR on knowledge graphs which predominately satisfy the assumptions of Faruqui et al. (2015), we extract a simple knowledge graph of lemmas and the connections between these lemmas that are annotated in WordNet. These connections are dominated by hypernymy and hyponymy (Table 7), which correlate with similarity, so we expect the baseline retrofitting method to perform well.

Results

As seen in Table 2, the increased flexibility of the FR framework does not degrade embedding quality even when this extra flexibility is not intuitively necessary. Here, we evaluate standard lexical metrics for word embeddings: word similarity and syntatic relations. For word similiarity tasks, the evaluation metric is the Spearman correlation between predicted and annotated similarities; for syntatic relation, the evaluation metric is the mean cosine similarity between the learned representation of the correct answer and the prediction by the vector offset method (Mikolov et al., 2013b). In contrast to our other experiments, here the only stochastic behavior is due to stochastic gradient descent training, not sampling of evaluation samples. Even though the WordNet knowledge graph largely satisfies the assumptions of the naïve retrofitting model, the flexible FR framework achieves sustained improvements for both word similarity datasets (WordSim-353; Finkelstein et al. (2001), Mturk-771[1], and MTurk-287and syntatic relations (Google Analogy Test Set[2]).

[1] http://www2.mta.ac.il/~gideon/mturk771.html

[2] http://download.tensorflow.org/data/questions-words.txt

Retrofitting Model	Word Similarity			Syntactic Relation
	WordSim-353	MTurk-771	MTurk-287	Google Analogy
None	0.512	0.538	0.671	0.772
FR-Identity	0.512	0.532	0.664	0.774
FR-Linear	**0.542**	**0.562**	**0.679**	**0.793**
FR-Neural	$\underline{0.516 \pm 0.001}$	$\underline{0.543 \pm 0.001}$	$\underline{0.676 \pm 0.001}$	$\underline{0.784 \pm 0.000}$

Table 2: Retrofitting to WordNet. Reported values are Spearman correlations for the word similarity tasks and mean cosine similarity for the syntatic relation task. These are deterministic evaluations, so the only source of stochasticity is the optimization of the FR-Neural model.

5.3 SNOMED-CT

SNOMED-CT is an ontology of clinical healthcare terms and concepts including diseases, treatments, anatomical terms, and many other types of entities. From the publicly available SNOMED-CT knowledge graph,[3] we extracted 327,001 entities and 3,809,639 edges of 169 different types (Table 8). To create distributional embeddings, we first link each SNOMED-CT concept to a set of Wikipedia articles by indexing the associated search terms in WikiData.[4] We aggregate each article set by the method of Arora et al. (2016), which performs TF-IDF weighted aggregation of pre-trained term embeddings to create sophisticated distributional embeddings of SNOMED-CT concepts. This creates a single 300-dimensional vector for each entity.

Results

As the SNOMED-CT ontology is dominated by synonymy-like relations, we expect the simple retrofitting methods to perform well. Nevertheless, we see minimal loss in link prediction performance by using the more flexible FR framework (Table 3). Our implementation supports the use of different function classes to represent different relation types; in practice, we recommend that users select function classes in accordance with relation semantics.

Retrofitting Model	'Has Finding Site' (113748/49070)	'Has Pathological Process' (19318/8124)	'Due to' (5042/2042)	'Cause of' (1166/376)
None	95.26 ± 0.01	98.79 ± 0.07	91.47 ± 0.88	79.61 ± 1.27
FR-Identity	$\underline{95.25 \pm 0.11}$	$\underline{99.09 \pm 0.11}$	$\mathbf{94.69 \pm 0.61}$	$\mathbf{86.67 \pm 1.27}$
FR-Linear	$\mathbf{95.35 \pm 0.01}$	$\mathbf{99.35 \pm 0.01}$	$\underline{93.50 \pm 0.46}$	$\underline{80.82 \pm 0.49}$
FR-Neural	95.22 ± 0.00	98.97 ± 0.22	91.70 ± 0.15	80.29 ± 0.80

Table 3: Retrofitting to SNOMED-CT. Reported values are mean and standard deviation of the link prediction accuracies over three experiments. The number of edges used for (training/testing) is shown below each edge type.

5.4 Roam Health Knowledge Graph

Finally, we investigate the utility of FR in the Roam Health Knowledge Graph (RHKG). The RHKG is a rich picture of the world of healthcare, with connections into numerous data sources: diverse medical ontologies, provider profiles and networks, product approvals and recalls, population health statistics, academic publications, financial data, clinical trial summaries and statistics, and many others. As of June 2, 2017 the RHKG contains 209,053,294 vertices, 1,021,163,726 edges, and 6,231,287,999 attributes. Here, we build an instance of the RHKG using only public data sources involving drugs and diseases.

[3] https://www.nlm.nih.gov/healthit/snomedct/index.html

[4] https://dumps.wikimedia.org/wikidatawiki/entities/

The structure of this knowledge graph is summarized in Table 9. In total, we select 48,649 disease–disease relations, 227,051 drug–drug relations, and 13,667 drug–disease relations used for retrofitting. A disjoint set of 11,306 drug–disease relations is reserved for evaluation.

In the RHKG, as in many industrial knowledge graphs, different distributional corpora are available for each type of entity. First, we mine 2.9M clinical texts for co-occurrence counts in physician notes. After counting co-occurrences, we perform a pointwise mutual information transform and ℓ_2 row normalization to generate embeddings for each entity. For drug embeddings, we supplement these embeddings with physician prescription habits. We extract prescription counts for each of 808,020 providers in the 2013 Centers for Medicare & Medicaid (CMS) dataset[5] and 837,629 providers in the 2014 CMS dataset. By aggregating prescriptions counts across provider specialty, we produce 201-dimensional distributional embeddings for each drug. Finally, we retrofit these distributional embeddings to the structure of the knowledge graph (excluding 'Treats' edges reserved for evaluation).

Results

As shown in Table 5, the FR framework significantly improves prediction of 'Treats' relations. We hypothesize that this is due to the separable nature of the graph; as seen in Figure 2, the FR retrofitting framework can learn Disease and Drug subgraphs that are nearly separable. In contrast, Identity retrofitting generates a single connected space and distorts the embeddings.

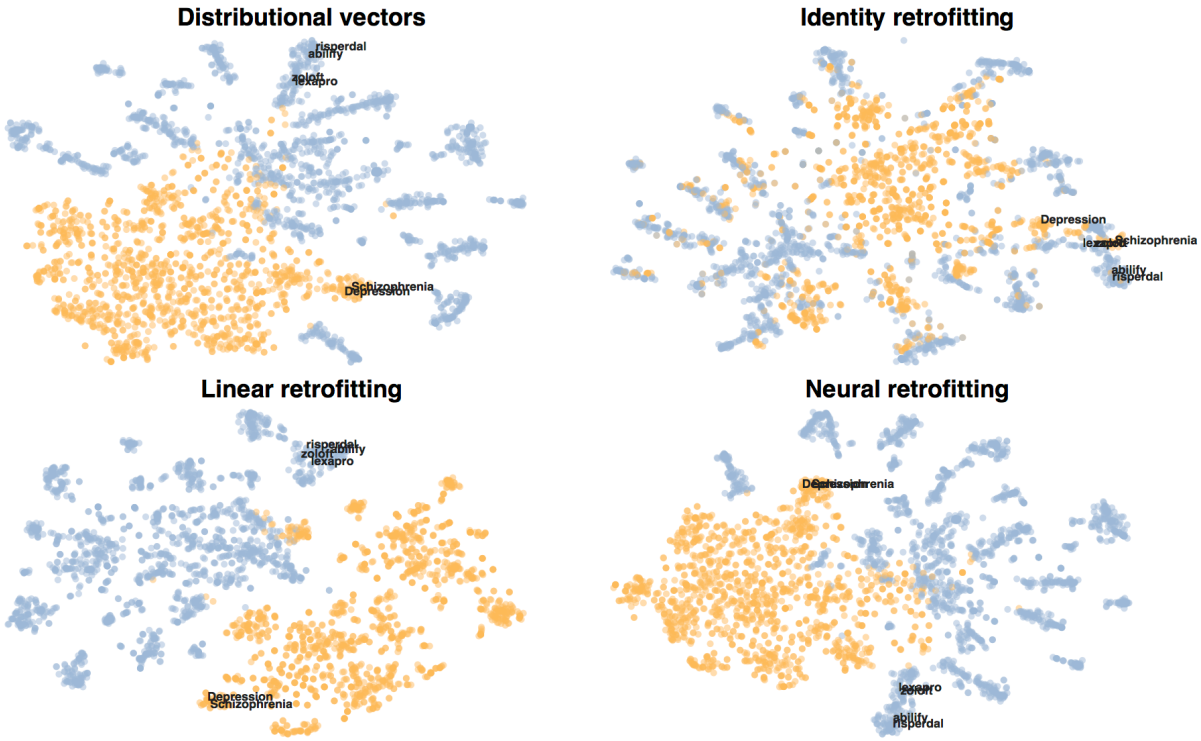

Figure 2: t-SNE Projections of the retrofitted embeddings of the drugs (blue) and diseases (orange) in the Roam Health Knowledge Graph, with selected annotations reflecting the 'Treats' relation. The distributional space strongly separates the two kinds of entity because their representations were learned in different ways. Identity retrofitting blurs this basic semantic distinction in order to make diseases and drugs in 'Treats' relations more similar. As Table 5 shows, the FR models achieve this same unification, but they need not distort the basic drug/disease distinction to do it.

We also investigate the predictions induced by the retrofitted representations. An interesting use of healthcare knowledge graphs is to predict drug *retargets*, that is, diseases for which there is no annotated treatment relationship with the drug but such a relationship may exist medically. As shown in Table 4,

[5] https://www.cms.gov/Research-Statistics-Data-and-Systems/Statistics-Trends-and-Reports/Medicare-Provider-Charge-Data/Part-D-Prescriber.html

Retrofitting Model	Drug	Disease Target	Model Score	Plausible
None	Naproxen	Ankylosing Spondylitis	0.98	Y
	Latanoprost	Superficial injury of ankle, foot and toes	0.96	N
	Pulmicort	Psoriasis, unspecified	0.96	Y
	Furosemide	Aneurysm of unspecified site	0.92	Y
	Desonide	Chlamydial lymphogranuloma (venereum)	0.92	N
FR-Identity	Latanoprost	Superficial injury of ankle, foot and toes	0.98	N
	Elixophyllin	Pneumonia in diseases classified elsewhere	0.94	Y
	Furosemide	Aneurysm of unspecified site	0.92	Y
	Oxistat	Mycosis fungoides	0.90	Y
	Trifluridine	Congenital Pneumonia	0.90	N
FR-Linear	Kenalog	Unspecified contact dermatitis	0.96	Y
	Kenalog	Pemphigus	0.96	Y
	Methyprednisolone Acetate	Nephrotic Syndrome	0.96	Y
	Furosemide	Aneurysm of unspecified site	0.94	Y
	Dexamethasone	Pemphigus	0.90	Y
FR-Neural	Onglyza	Type 2 diabetes mellitus	0.98	Y
	Pradaxa	Essential (primary) hypertension	0.96	Y
	Oxytocin	Pauciarticular juvenile rheumatoid arthritis	0.94	Y
	Terbutaline sulfate	HIV 2 as the cause of diseases classified elsewhere	0.94	N
	Lipitor	Cerebral infarction	0.92	Y

Table 4: Highest confidence drug targets that were not annotated in the Roam Health Knowledge Graph.

Retrofitting Model	'Treats' (9152/2490)
None	72.02 ± 0.50
FR-Identity	72.93 ± 0.82
FR-Linear	$\mathbf{84.22 \pm 0.82}$
FR-Neural	$\underline{73.52 \pm 0.89}$

Table 5: Drug-Disease Link Prediction Accuracies.

the top retargets predicted by the linear retrofitting model are all medically plausible. In particular, the model confidently predicts that Kenalog would treat contact dermatitis, an effect also found in a clinical trial (Usatine and Riojas, 2010).The second most confident prediction of drug retargets was that Kenalog can treat pemphigus, which is indicated on Kenalog's drug label,[6] but was not previously included in the knowledge graph. The third prediction was that methyprednisolone acetate would treat nephrotic syndrome, which is reasonable as the drug is now labelled to treat idiopathic nephrotic syndrome.[7] Interestingly, several models predict that furosemide treats "aneurysm of unspecified site", a relationship not indicated on the drug label[8], though furosemide has been observed to reduce intracranial pressure (Samson and Beyer Jr, 1982), a key factor in brain aneurysms. Finally, both the distributional data and the embeddings produced by the baseline identity retrofitting model make the nonsensical prediction that Latanoprost, a medication used to treat intraocular pressure, would also treat superficial ankle and foot injuries.

The accuracy of the predictions from the more complex models underscores the utility of the new framework for retrofitting distributional embeddings to knowledge graphs with relations that do not imply similarity.

[6] https://www.accessdata.fda.gov/drugsatfda_docs/label/2014/014901s042lbledt.pdf

[7] https://dailymed.nlm.nih.gov/dailymed/fda/fdaDrugXsl.cfm?setid=978b8416-2e88-4816-8a37-bb20b9af4b1d

[8] https://dailymed.nlm.nih.gov/dailymed/drugInfo.cfm?setid=eadfe464-720b-4dcd-a0d8-45dba706bd33

6 Conclusions and Future Work

We have presented *Functional Retrofitting*, a new framework for *post-hoc* retrofitting of entity embeddings to the structure of a knowledge graph. By explicitly modeling pairwise relations, this framework allows users to encode, learn, and extract information about relation semantics while simultaneously updating entity representations. This framework extends the popular concept of retrofitting to knowledge graphs with diverse entity and relation types. Functional Retrofitting is especially beneficial for graphs in which distinct distributional corpora are available for different entity classes, but it loses no accuracy when applied to simpler knowledge graphs. Finally, we are interested in the possibility of improvements to the optimization procedure outlined in this paper, including dynamic updates of the β and α parameters to increase trust in the graph structure while the relation functions are learned.

Acknowledgements

We would like to thank Adam Foster, Ben Peloquin, JJ Plecs, and Will Hamilton for insightful comments, and anonymous reviewers for constructive criticism.

References

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2016. A simple but tough-to-beat baseline for sentence embedtdings. *International Conference on Learning Representations*.

Collin F Baker, Charles J Fillmore, and John B Lowe. 1998. The berkeley framenet project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1*, pages 86–90. Association for Computational Linguistics.

Antoine Bordes, Xavier Glorot, Jason Weston, and Yoshua Bengio. 2012. Joint learning of words and meaning representations for open-text semantic parsing. In *JMLR W&CP: Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics (AISTATS 2012)*.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, pages 2787–2795.

Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard Hovy, and Noah A. Smith. 2015. Retrofitting word vectors to semantic lexicons. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1606–1615. Association for Computational Linguistics.

Manaal Faruqui, Ryan McDonald, and Radu Soricut. 2016. Morpho-syntactic lexicon generation using graph-based semi-supervised learning. *Transactions of the Association of Computational Linguistics*, 4(1):1–16.

Christiane Fellbaum. 2005. Wordnet and wordnets. In Keith Brown et al., editor, *Encyclopedia of Language and Linguistics*, page 665670. Oxford: Elsevier, second edition.

Charles J Fillmore, Christopher R Johnson, and Miriam RL Petruck. 2003. Background to framenet. *International journal of lexicography*, 16(3):235–250.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2001. Placing search in context: The concept revisited. In *Proceedings of the 10th international conference on World Wide Web*, pages 406–414. ACM.

William L Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *arXiv preprint arXiv:1706.02216*.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *AAAI*, pages 2181–2187.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013a. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013b. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, volume 13, pages 746–751.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Nikola Mrkšic, Diarmuid OSéaghdha, Blaise Thomson, Milica Gašic, Lina Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. Counter-fitting word vectors to linguistic constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 142–148.

Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. 2016. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1):11–33.

Duke Samson and Chester W Beyer Jr. 1982. Furosemide in the intraoperative reduction of intracranial pressure in the patient with subarachnoid hemorrhage. *Neurosurgery*, 10(2):167–169.

Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. 2013. Reasoning with neural tensor networks for knowledge base completion. In *Advances in neural information processing systems*, pages 926–934.

Richard P Usatine and Marcela Riojas. 2010. Diagnosis and management of contact dermatitis. *American family physician*, 82(3):249–255.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*.

A Structure of Knowledge Graphs

A.1 FrameNet

The structure of the FrameNet (Baker et al., 1998; Fillmore et al., 2003) knowledge graph is shown in Table 6.

Entity Type	**Count**	**W2V Count**
Token	13572	12167
Frame	1221	464

Edge Type	**Connects**	**Count**
Frame	Token → Frame	13572
Lexical_Unit	Frame → Token	13572
Inheritance	Frame → Frame	1562
Using	Frame → Frame	1110
ReFraming_Mapping	Frame → Frame	428
Persepctive_on	Frame → Frame	242
Precedes	Frame → Frame	178
See_also	Frame → Frame	172
Causative_of	Frame → Frame	120
Inchoative_of	Frame → Frame	38
Metaphor	Frame → Frame	8

Table 6: Structure of the FrameNet knowledge graph.

A.2 WordNet

The structure of the WordNet knowledge graph (Miller, 1995) is shown in Table 7.

Entity Type	Count	W2V Count
Lemma	206978	115635

Edge Type	Count
Hypernym	136,235
Hyponym	136,235
Derivationally Related Form	60,250
Antonym	5,922
Pertainym	5,573
Usage Domain	69

Table 7: Structure of the WordNet knowledge graph.

A.3 SNOMED-CT

The structure of the knowledge graph extracted from the SNOMED-CT ontology is shown in Table 8.

A.4 Roam Health Knowledge Graph

The structure of the extracted subgraph of the RHKG is summarized in Table 9. A disjoint set of 11,306 drug–disease relations is reserved for evaluation.

Edge Type	Count	Edge Type	Count
associated_clinical_finding	493258	child	242130
has_finding_site	205263	has_method	200507
has_associated_morphology	169778	has_procedure_site	79171
has_causative_agent	69284	interprets	67900
has_active_ingredient	58976	part_of	47776
has_direct_procedure_site	45693	mapped_to	37287
same_as	30670	has_pathological_process	23641
has_dose_form	23259	has_intent	22845
causative_agent_of	19833	finding_site_of	19525
has_direct_morphology	18380	has_direct_substance	16913
has_component	15597	has_indirect_procedure_site	15596
occurs_in	14003	possibly_equivalent_to	13459
has_finding_method	12754	active_ingredient_of	12423
has_definitional_manifestation	11788	has_direct_device	11223
is_interpreted_by	10908	has_interpretation	10077
procedure_site_of	9559	occurs_after	7825
has_temporal_context	7786	associated_morphology_of	7524
has_subject_relationship_context	7465	has_part	6851
uses_device	6407	associated_with	6399
has_measured_component	6353	uses	6221
has_associated_finding	6205	has_focus	6122
uses_substance	5474	component_of	5256
temporally_follows	5029	due_to	4884
has_finding_context	4883	direct_procedure_site_of	4252
has_specimen	3767	replaces	3726
has_laterality	3641	associated_finding_of	3432
has_associated_procedure	3397	has_clinical_course	3309
has_course	3241	has_procedure_context	2945
has_approach	2808	measured_component_of	2741
has_access	2660	has_specimen_source_topography	2457
has_finding_informer	2229	has_onset	2168
has_priority	1854	mth_xml_form_of	1794
mth_plain_text_form_of	1794	mth_has_xml_form	1794
mth_has_plain_text_form	1794	direct_substance_of	1783
focus_of	1680	indirect_procedure_site_of	1662
has_revision_status	1599	uses_access_device	1587
has_access_instrument	1518	direct_device_of	1434
has_indirect_morphology	1426	associated_procedure_of	1320
has_specimen_procedure	1309	has_communication_with_wound	1155
cause_of	1121	has_extent	1082
has_specimen_substance	1030	method_of	921
has_procedure_device	770	uses_energy	753
has_procedure_morphology	752	has_surgical_approach	697
dose_form_of	676	direct_morphology_of	673
referred_to_by	667	has_associated_etiologic_finding	656
used_by	608	priority_of	586
specimen_source_topography_of	584	occurs_before	574
specimen_procedure_of	555	has_severity	525
device_used_by	525	substance_used_by	507
definitional_manifestation_of	436	temporally_followed_by	406
has_specimen_source_identity	327	has_property	282
has_instrumentation	274	has_subject_of_information	272
has_specimen_source_morphology	251	access_instrument_of	226
has_scale_type	206	specimen_substance_of	171
has_episodicity	168	has_route_of_administration	143
has_recipient_category	143	associated_etiologic_finding_of	143
specimen_of	134	approach_of	125
subject_relationship_context_of	115	has_indirect_device	114
interpretation_of	109	procedure_device_of	107
course_of	106	indirect_morphology_of	10

Table 8: Structure of the SNOMED-CT knowledge graph.

Entity Type	Count
Drug	223,019
Disease	95,559

Edge Type	Connects	Count
Ingredient Of	Drug → Drug	49,218
Has Ingredient	Drug → Drug	49,208
Is A	Drug → Drug	28,297
Has Descendent	Disease → Disease	22,344
Treats	Drug → Disease	19,374
Has Active Ingredient	Drug → Drug	18,422
Has Child	Disease → Disease	18,066
Active Ingredient Of	Drug → Drug	17,175
Has TradeName	Drug → Drug	11,783
TradeName Of	Drug → Drug	11,783
Inverse Is A	Drug → Drug	10,369
Has Symptom	Disease → Disease	7,892
Part Of	Drug → Drug	6,882
Has Part	Drug → Drug	6,624
Same As	Drug → Drug	5,882
Precise Ingredient Of	Drug → Drug	3,562
Has Precise Ingredient	Drug → Drug	3,562
Possibly Equivalent To	Drug → Drug	1,233
Causative Agent of	Drug → Drug	1,070
Has Form	Drug → Drug	602
Form of	Drug → Drug	602
Component of	Drug → Drug	436
Includes	Disease → Disease	347
Has Dose Form	Drug → Drug	138

Table 9: Structure of the subgraph of the Roam Health Knowledge Graph.