Holographic Embeddings of Knowledge Graphs

Maximilian Nickel\textsuperscript{1,2} and Lorenzo Rosasco\textsuperscript{1,2,3} and Tomaso Poggio\textsuperscript{1}

\textsuperscript{1}Laboratory for Computational and Statistical Learning and Center for Brains, Minds and Machines Massachusetts Institute of Technology, Cambridge, MA
\textsuperscript{2}Istituto Italiano di Tecnologia, Genova, Italy
\textsuperscript{3}DIBRIS, Universita Degli Studi Di Genova, Italy

Abstract

Learning embeddings of entities and relations is an efficient and versatile method to perform machine learning on relational data such as knowledge graphs. In this work, we propose holographic embeddings (H\textsuperscript{o}E) to learn compositional vector space representations of entire knowledge graphs. The proposed method is related to holographic models of associative memory in that it employs circular correlation to create compositional representations. By using correlation as the compositional operator, H\textsuperscript{o}E can capture rich interactions but simultaneously remains efficient to compute, easy to train, and scalable to very large datasets. Experimentally, we show that holographic embeddings are able to outperform state-of-the-art methods for link prediction on knowledge graphs and relational learning benchmark datasets.

Introduction

Relations are a key concept in artificial intelligence and cognitive science. Many of the structures that humans impose on the world, such as logical reasoning, analogies, or taxonomies, are based on entities, concepts and their relationships. Hence, learning from and with relational knowledge representations has long been considered an important task in artificial intelligence (see e.g., (Getoor and Taskar 2007; Muggleton 1991; Gentner 1983; Kemp et al. 2006; Xu et al. 2006; Richardson and Domingos 2006)). In this work we are concerned with learning from knowledge graphs (KGs), i.e., knowledge bases which model facts as instances of binary relations (e.g., bornIn(BarackObama, Hawaii)). This form of knowledge representation can be interpreted as a multigraph, where entities correspond to nodes, facts correspond to typed edges, and the type of an edge indicates the kind of the relation. Modern knowledge graphs such as YAGO (Suchanek, Kasneci, and Weikum 2007), DBpedia (Auer et al. 2007), and Freebase (Bollacker et al. 2008) contain billions of facts about millions of entities and have found important applications in question answering, structured search, and digital assistants. Recently, vector space embeddings of knowledge graphs have received considerable attention, as they can be used to create statistical models of entire KGs, i.e., to predict the probability of any possible relation instance (edge) in the graph. Such models can be used to derive new knowledge from known facts (link prediction), to disambiguate entities (entity resolution), to extract taxonomies, and for probabilistic question answering (see e.g., (Nickel, Tresp, and Kriegel 2011; Bordes et al. 2013; Krompaß, Nickel, and Tresp 2014)). Furthermore, embeddings of KGs have been used to support machine reading and to assess the trustworthiness of web sites (Dong et al. 2014; 2015). However, existing embedding models that can capture rich interactions in relational data are often limited in their scalability. Vice versa, models that can be computed efficiently are often considerably less expressive. In this work, we approach learning from KGs within the framework of compositional vector space models. We introduce holographic embeddings (H\textsuperscript{o}E) which use the circular correlation of entity embeddings (vector representations) to create compositional representations of binary relational data. By using correlation as the compositional operator H\textsuperscript{o}E can capture rich interactions but simultaneously remains efficient to compute, easy to train, and scalable to very large datasets. As we will show experimentally, H\textsuperscript{o}E is able to outperform state-of-the-art embedding models on various benchmark datasets for learning from KGs. Compositional vector space models have also been considered in cognitive science and natural language processing, e.g., to model symbolic structures, to represent the semantic meaning of phrases, and as models for associative memory (see e.g., (Smolensky 1990; Plate 1995; Mitchell and Lapata 2008; Socher et al. 2012)). In this work, we do not only draw inspiration from these models, but we will also highlight the connections of H\textsuperscript{o}E to holographic models of associative memory.

Compositional Representations

In this section we introduce compositional vector space models for KGs, the general learning setting, and related work.

Let $\mathcal{E}$ denote the set of all entities and $\mathcal{P}$ the set of all relation types (predicates) in a domain. A binary relation is a subset $R_p \subseteq \mathcal{E} \times \mathcal{E}$ of all pairs of entities (i.e., those pairs which are in a relation of type $p$). Higher-arity relations are defined analogously. The characteristic function $\phi_p : \mathcal{E} \times \mathcal{E} \rightarrow \{\pm 1\}$ of a relation $R_p$ indicates for each possible pair of entities whether they are part of $R_p$. We will denote (possible) relation instances as $R_p(s, o)$, where $s, o \in \mathcal{E}$ denote the first and second argument of the asymmetric relation $R_p$. We will
where tensor product models represent pairs of entities via interactions between the features of a, b

\[ \Theta \]

following compositional operators:

For a review of machine learning global dependencies in the data, and to enable the desired allows to propagate information between triples, to capture relations are learned jointly in eqs. (2) and (3), this property and object. Since the representations of all entities and relations

\[ \Theta \]

to rank the probability of existing triples higher than the probability of non-existing ones. Here, \( D^+, D^- \) denote the set of existing and non-existing triples and \( \gamma > 0 \) specifies the width of the margin (Bordes et al. 2011).

An important property of compositional models is that the mapping from the tuple representation

\[ \psi \]

specifies

\[ \Theta \]

parameters. However, this approach can only model symmetric relations and is not suitable to model general knowledge graphs as \( e^s \sim R_p e^o \) for diagonal \( R_p \).

Concatenation, Projection, and Non-Linearly

Another way to compute composite representations is via concatenation, projection and subsequent application of a non-linear function. Let \( \odot : \mathbb{R}^{d_x} \times \mathbb{R}^{d_z} \rightarrow \mathbb{R}^{d_x+d_z} \) denote concatenation and \( \psi : \mathbb{R} \rightarrow \mathbb{R} \) be a non-linear function such as \( \psi \). The composite tuple representation is then given by

\[ \Theta \]

to rank the probability of existing triples higher than the probability of non-existing ones. Here, \( D^+, D^- \) denote the set of existing and non-existing triples and \( \gamma > 0 \) specifies the width of the margin (Bordes et al. 2011).

An important property of compositional models is that the mapping from the tuple representation

\[ \psi \]

specifies

\[ \Theta \]

parameters. However, this approach can only model symmetric relations and is not suitable to model general knowledge graphs as \( e^s \sim R_p e^o \) for diagonal \( R_p \).

Concatenation, Projection, and Non-Linearly

Another way to compute composite representations is via concatenation, projection and subsequent application of a non-linear function. Let \( \odot : \mathbb{R}^{d_x} \times \mathbb{R}^{d_z} \rightarrow \mathbb{R}^{d_x+d_z} \) denote concatenation and \( \psi : \mathbb{R} \rightarrow \mathbb{R} \) be a non-linear function such as \( \psi \). The composite tuple representation is then given by

\[ \Theta \]

to rank the probability of existing triples higher than the probability of non-existing ones. Here, \( D^+, D^- \) denote the set of existing and non-existing triples and \( \gamma > 0 \) specifies the width of the margin (Bordes et al. 2011).

An important property of compositional models is that the mapping from the tuple representation

\[ \psi \]

specifies

\[ \Theta \]

parameters. However, this approach can only model symmetric relations and is not suitable to model general knowledge graphs as \( e^s \sim R_p e^o \) for diagonal \( R_p \).
A major appeal of TRANS E is that it requires very few parameters and moreover is easy to train. However, this simplicity comes also at the cost of modeling power. Wang et al. (2014) and Lin et al. (2015) proposed TRANS H and TRANS R respectively, to improve the performance of TRANS E on 1-to-N, N-to-1, and N-to-N relations. Unfortunately, these models lose the simplicity and efficiency of TRANS E.

### Holographic Embeddings

In this section, we propose a novel compositional model for KGs and relational data. To combine the expressive power of the tensor product with the efficiency and simplicity of TRANS E, we use the circular correlation of vectors to represent pairs of entities, i.e., we use the compositional operator:

\[ a \circ b = a \star b, \tag{7} \]

where \( \star : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}^d \) denotes circular correlation:

\[ [a \star b]_k = \sum_{i=0}^{d-1} a_i b_{(k+i) \mod d}. \tag{8} \]

Hence, we model the probability of a triple as

\[ \Pr(\phi_p(s,o)) = 1(\Theta) = \sigma(r_p^\top (e_s \star e_o)). \tag{9} \]

Due to its connections to holographic models of associative memory (which we will discuss in the next section) we refer to eq. (9) as holographic embeddings (HOLE) of KGs.

As a compositional operator, circular correlation can be interpreted as a compression of the tensor product. While the tensor product assigns a separate component \( c_{ij} = a_i b_j \) for each pairwise interaction of entity features, in correlation each component corresponds to a sum over a fixed partition of pairwise interactions (see also fig. 2). Intuitively, a feature in the tuple representation is “on” if at least one partition of subject-object-interactions is on. This form of compression can be very effective since it allows to share weights in \( r_p \) for semantically similar interactions. For example, to model relational patterns in the partyOf relation, it might be sufficient to know whether subject and object are a liberal person and liberal party OR if they are a conservative person and conservative party. These interactions can then be grouped in the same partition. Additionally, it is typically the case that only a subset of all possible interactions of latent features are relevant to model relational patterns. Irrelevant interactions can then be grouped in the same partitions and collectively be assigned a small weight in \( r_p \). Please note that the partitioning is not learned but fixed in advance through the correlation operator. This is possible because the entity representations are learned and the latent features can thus be “assigned” to the best partition during learning.

Compared to the tensor product, circular correlation has the important advantage that it does not increase the dimensionality of the composite representation (see also fig. 1b). The memory complexity of the tuple representation is therefore linear in the dimensionality \( d \) of the entity representations. Moreover, the runtime complexity is quasilinear (loglinear) in \( d \), as circular correlation can be computed via

\[ a \star b = F^{-1} \left( F(a) \circ F(b) \right) \]

where \( F(\cdot) \) and \( F^{-1}(\cdot) \) denote the fast Fourier transform (FFT) and its inverse. \( \ast \) denotes the complex conjugate of...
\[ c = a \times b \]

\[ c_0 = a_0 b_0 + a_1 b_1 + a_2 b_2 \]
\[ c_1 = a_0 b_2 + a_1 b_0 + a_2 b_1 \]
\[ c_2 = a_0 b_1 + a_1 b_2 + a_2 b_0 \]

\[ e = a \times b \]

Figure 2: Circular correlation as compression of the tensor product. Arrows indicate summation patterns, nodes indicate elements in the tensor product. Adapted from (Plate 1995).

\[ x \in \mathbb{C}^d, \text{ and } \odot \text{ denotes the Hadamard (entrywise) product.} \]

The computational complexity of the FFT is \( \mathcal{O}(d \log d) \). Table 1a summarizes the improvements of circular correlation over the tensor product. Table 1b lists the memory complexity of HoTeLi in comparison to other embedding models.

Circular convolution \( *: \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}^d \) is an operation that is closely related to circular correlation and defined as

\[ [a \times b]_k = \sum_{i=0}^{d-1} a_i b_{(k-i) \mod d}. \]  

In comparison to convolution, correlation has two main advantages when used as a compositional operator:

Non Commutative Correlation, unlike convolution, is not commutative, i.e., \( a \times b \neq b \times a \). Non-commutativity is necessary to model asymmetric relations (directed graphs) with compositional representations.

Similiarity Component In the correlation \( a \times b \), a single component \( [a \times b]_0 = \sum_i a_i b_i \) corresponds to the dot product \( \langle a, b \rangle \). The existence of such a component can be helpful to model relations in which the similarity of entities is important. No such component exists in the convolution \( a * b \) (see also fig. 1 in the supplementary material).

To compute the representations for entities and relations, we minimize either eq. (2) or (3) via stochastic gradient descent (SGD). Let \( \theta \in \{ e_i \}_{i=1}^n \cup \{ r_k \}_{k=1}^m \) denote the embedding of a single entity or relation and let \( f_{spo} = \sigma(\mathbf{r}_p \langle \mathbf{e}_s, \mathbf{e}_o \rangle) \). The gradients of eq. (9) are then given by

\[ \frac{\partial f_{spo}}{\partial \theta} = \frac{\partial f_{spo}}{\partial \eta_{spo}} \frac{\partial \eta_{spo}}{\partial \theta}. \]

where

\[ \frac{\partial \eta_{spo}}{\partial \mathbf{r}_p} = \mathbf{e}_s \cdot \mathbf{e}_o, \quad \frac{\partial \eta_{spo}}{\partial \mathbf{e}_s} = \mathbf{r}_p \cdot \mathbf{e}_o, \quad \frac{\partial \eta_{spo}}{\partial \mathbf{e}_o} = \mathbf{r}_p \cdot \mathbf{e}_s. \]

(11)

The partial gradients in eq. (11) follow directly from

\[ \mathbf{r}_p^\top (\mathbf{e}_s \cdot \mathbf{e}_o) = \mathbf{r}_s^\top (\mathbf{r}_p \cdot \mathbf{e}_o) = \mathbf{r}_o^\top (\mathbf{r}_p \cdot \mathbf{e}_s) \]

(12)

and standard vector calculus. Equation (12) can be derived as follows: First we rewrite correlation in terms of convolution:

\[ a \times b = \tilde{a} \ast \tilde{b} \]

where \( \tilde{a} \) denotes the involution of \( a \), meaning that \( \tilde{a} \) is the mirror image of \( a \) such that \( \tilde{a}_i = a_{-i \mod d} \) (Schönenmann 1987, eq. 2.4). Equation (12) follows then from the following identities in convolution algebra (Plate 1995):

\[ \mathbf{c}^\top (\tilde{a} \ast b) = \mathbf{a}^\top (\mathbf{c} \ast b); \quad \mathbf{c}^\top (\tilde{a} \ast b) = \mathbf{b}^\top (\mathbf{a} \ast \mathbf{c}). \]

Similar to correlation, the circular convolution in eq. (11) can be computed efficiently via \( a \ast b = \mathcal{F}^{-1}(\mathcal{F}(a) \odot \mathcal{F}(b)) \).

**Associative Memory**

In this section we outline the connections of eq. (9) and eq. (11) to holographic models of associative memory. Such models employ a sequence of convolutions and correlations as used in holography to store and retrieve information (e.g., see (Gabor 1969; Poggio 1973)). In particular, holographic reduced representations (Plate 1995) store the association of \( a \) with \( b \) via their circular convolution

\[ m = a \times b, \]

and retrieve the item associated with \( a \) from \( m \) via

\[ b' \approx a \times m = b \ast (a \ast a) \]

If \( a \ast a \approx \delta \) (the identity element of convolution), it holds that \( b \approx b' \) and we can retrieve a noisy version of \( b \). For denoising, we can pass the retrieved vector through a clean-up memory, which returns the stored item with the highest similarity to the item retrieved from \( m \). For instance, if \( \|a\| = \|b\| = 1 \), we can perform the clean-up via

\[ b = \arg \max_{\mathbf{b}} \mathbf{b}^\top (a \times m) \]

(13)

Multiple elements are stored in \( m \) via superposition:

\[ m = \sum_i a_i \times b_i. \]

Hence, \( m \) acts in this scheme as a memory that stores associations between vectors which are stored and retrieved using circular convolution and correlation.

Consider now the following model of associative memory for relational data: Let \( S_o = \{(s, p) | \phi_p(s, o) = 1 \} \) be the set of all subject-predicate indices for which the relation \( R_p(s, o) \) is true. Next, we store these existing relations via convolution and superposition in the representation \( e_o \):

\[ e_o = \sum_{(s, p) \in S_o} r_p \ast e_s \]

(14)

In this scheme, the compositional representation \( e_s \times e_o \) of eq. (7) would be analogous to retrieving the stored predicates \( p \) that exist between \( s \) and \( o \). Similarly, computing \( \sigma(r_p \langle e_s, e_o \rangle) \) as in eq. (9) is analogous to computing the probability that \( r_p \) is included in the retrieved relations, i.e., that we have seen the triple \( R_p(s, o) \). The norm constraints of eq. (13) can either be enforced directly (by projection the embeddings onto the unit circle) or through the regularization of the embeddings (which is equivalent to \( \|e_i\| \leq C_e, \||r_k|| \leq C_r \)), where \( C_e, C_r \) depend on the regularization parameter.)
An important difference of HoLE to associative memory is that it does not only memorize, but it generalizes in a well-defined way: In associative memory we are given the embeddings and store the associations directly, typically via Hebbian learning (e.g., see eq. (14)). In HoLE, we do not simply store the associations, but instead learn the embeddings that best explain the observed data. By iterating over $D$ with SGD, we update the embeddings of the objects via

$$ e_o^{t+1} \leftarrow e_o - \mu \frac{\partial L}{\partial f} \frac{\partial f}{\partial e} (r_p \circ e_s), \quad (15) $$

where $\mu$ denotes the learning rate. Please note that eq. (15) is analogous to the association of predicate and subject in holographic associative memory. Hence, we can interpret $\mu$ as adapting the “memory” $e_o$, such that the retrieval of the observed facts is improved. The same analogy holds for the updates of $e_s$ and $r_p$, however with the roles of correlation and convolution in storage and retrieval reversed. Moreover, in minimizing eq. (2) via eq. (15), we are estimating a probability distribution over possible states of the knowledge graph which allows us to predict the probability of any possible triple in the graph (Nickel et al. 2015).

Experiments

Knowledge Graphs

To evaluate its performance for link prediction on knowledge graphs, we compared HoLE to state-of-the-art models on two commonly used benchmark datasets for this task:

**WN18** WordNet is a KG that groups words into synonyms and provides lexical relationships between words. The WN18 dataset consists of a subset of WordNet, containing 40,943 entities, 18 relation types, and 151,442 triples.

**FB15k** Freebase is a large knowledge graph that stores general facts about the world (e.g., harvested from Wikipedia, MusicBrainz, etc.). The FB15k dataset consists of a subset of Freebase, containing 14,951 entities, 1345 relation types, and 592,213 triples.

For both datasets we used the fixed training-, validation-, and test-splits provided by Bordes et al. (2013). As baseline methods, we used RESCAL, TRANS E, TRANS R, and ER-MLP. To facilitate a fair comparison we reimplemented all models and used the identical loss and optimization method for training, i.e., SGD with AdaGrad (Duchi, Hazan, and Singer 2011) and the ranking loss of eq. (3). This improved the results of TransE and RESCAL significantly on both datasets compared to results reported by Bordes et al. (2013).\(^2\)

Following Bordes et al. (2013), we generated negative relation instances for training by corrupting positive triples and used the following evaluation protocol: For each true triple $R_p(s, o)$ in the test set, we replace the subject $s$ with each entity $s' \in \mathcal{E}$, compute the score for $R_p(s', o)$, and rank all these instances by their scores in decreasing order. Since there can exist multiple true triples in the test set for a given predicate-object pair, we remove all instances from the ranking where $R_p(s', o) = 1$ and $s \neq s'$, i.e., we consider only the ranking of the test instance among all wrong instances (which corresponds to the “Filtered” setting in Bordes et al. (2013)). We then repeat this procedure by replacing the object $o$. To measure the quality of the ranking, we use the mean reciprocal rank (MRR) which is commonly used in information retrieval and in contrast to mean rank is less sensitive to outliers. In addition to MRR, we report the ratio in which $R_p(s, o)$ occurs within the first $n$ results (denoted by “Hits at n”). We optimized the hyperparameters of all models via extensive grid search and selected the model with the best filtered MRR score on the validation set. The results of these experiments are shown in table 2a. It can be seen that HoLE is able to outperform the considered baseline methods significantly and consistently on both datasets. For instance, TRANS E and TRANS R rank the test instance only in 11.5% and 33.5% of the cases as the most likely triple in WN18 (Hits at 1). In contrast, HoLE ranks the test instance in 93.0% of the cases as the most likely instance. While less pronounced, similar results can be observed on FB15k. In table 1b, we report the dimensionality $d$ and the resulting number of parameters of the selected models. It can be seen that HoLE is far more efficient in the number of parameters compared to the tensor product model RESCAL. Although the dimensionality $d$ of the HoLE embedding is larger than RESCAL’s (what is to be expected due to the compressive effect of correlation), the overall number of parameters is significantly reduced as its memory complexity depends only linearly on $d$. Also, HoLE is typically very fast to compute. On standard hardware (Intel Core(TM) i7U 2.1GHz) the runtime is $d = 150$ (as used in the experiments) the runtime to compute the probability of a single triple is around $40 \mu s$. To compute all embeddings, a single epoch on WN18 takes around 11s (earlier epochs are slower since more examples violate the margin). Typically, we need 200-500 epochs (depending on the dataset) to arrive at the best estimates for the embeddings.

Relational Learning

We have shown that HoLE can predict triples successfully in knowledge graphs. In additional experiments, we wanted to test the relational learning capabilities of the compositional representation. For this purpose, we used the countries dataset of Bouchard, Singh, and Trouillon (2015), which consists of 244 countries, 22 subregions (e.g., Southern Africa, Western Europe) and 5 regions (e.g., Africa, Americas). Each country is located in exactly one region and subregion, each subregion is located in exactly one region, and each country can have a number of other countries as neighbors. From the raw data we created a relational representation with two predicates: locatedIn($c_1, c_2$) and neighborOf($c_1, c_2$). The task in the experiment was to predict locatedIn($c, r$) instances, where $c$ ranges over all countries and $r$ over all regions in the data. The evaluation protocol was the following: First, we split all countries randomly in train (80%), validation (10%), and test (10%) set, such that for each country in the test set there is at least one neighbor in the training set. Next, we removed triples from the test and validation set in three different settings:

- **S1** In the basic setting we only set locatedIn($c, r$) to missing for countries in the test/valid. set. In this setting, the correct

\(^2\)TRANS E in its original implementation used SGD without AdaGrad. RESCAL used the least-squares loss and ALS updates.
relations can be predicted from patterns of the form:
\[
\text{locatedIn}(c, s) \land \text{locatedIn}(s, r) \Rightarrow \text{locatedIn}(c, r)
\]
where \(s\) refers to the country’s subregion.

S2) In addition to the triples of S1, we set \(\text{locatedIn}(c, s)\) to missing for all countries \(c\) in the test/valid. set and all subregions \(s\) in the data. In this setting, the correct triples can be predicted from:
\[
\text{neighborOf}(c_1, c_2) \land \text{locatedIn}(c_2, r) \Rightarrow \text{locatedIn}(c_1, r)
\]
This is a harder task than S1, since a country can have multiple neighbors and these can be in different regions.

S3) In addition to the triples of S1 and S2 we set \(\text{locatedIn}(n, r)\) to missing for all neighbors \(n\) of all countries in the test/valid. set and all regions \(r\) in the data. In this setting, the correct triples can be predicted from:
\[
\text{neighborOf}(c_1, c_2) \land \text{locatedIn}(c_2, s) \land \\
\text{locatedIn}(s, r) \Rightarrow \text{locatedIn}(c_1, r)
\]
This is the most difficult task, as it not only involves the \(\text{neighborOf}\) relation, but also a path of length 3.

See fig. 3 for an illustration of the data structure and the test settings. We measured the prediction quality via the area under the precision-recall curve (AUC-PR). The results of the experiments are shown in table 2b. It can be seen that HOLE is very successful in these learning tasks. For S1, the missing triples are predicted nearly perfectly. Moreover, even for the most difficult task S3, HOLE achieves very good results, especially since not every country’s region can be predicted from its neighbors (e.g., islands have no neighbors). The poorer results of RESCAL and ER-MLP can likely be explained with overfitting (although the models are regularized), since the difference to HOLE is reduced when the hyperparameters are optimized on the test set instead of the validation set. We observed similar results as in this experiment on commonly used benchmark datasets for statistical relational learning. Due to space constraints, we report these experiments in the supplementary material.

**Conclusion and Future Work**

In this work we proposed HOLE, a compositional vector space model for knowledge graphs that is based on the circular correlation of vectors. An attractive property of circular correlation in this context is that it creates fixed-width representations, meaning that the compositional representation has the same dimensionality as the representation of its constituents. In HOLE, we exploited this property to create a compositional model that can capture rich interactions in relational data but simultaneously remains efficient to compute, easy to train, and very scalable. Experimentally we showed that HOLE provides state-of-the-art performance on a variety of benchmark datasets and that it can model complex relational patterns while being very economical in the number of its parameters. Moreover, we highlighted connections of HOLE to holographic models of associative memory and discussed how it can be interpreted in this context. This creates not only a link between relational learning and associative memory, but also allows for principled ways to query the model, for instance in question answering. In future work we plan to further exploit the fixed-width representations of holographic embeddings in complex scenarios, since they are especially suitable to model higher-arity relations (e.g., \(\text{taughtAt}(\text{John, AI, MIT})\)) and facts about facts (e.g., \(\text{believes}(\text{John, loves(\text{Tom, Mary})})\)).

**Acknowledgments & Reproducibility**

This material is based upon work supported by the Center for Brains, Minds and Machines (CBMM), funded by NSF STC award CCF-1231216. The code for models and experiments used in this paper is available at https://github.com/mnick/holographic-embeddings.