Compositional Vector Space Models for Knowledge Base Completion

Arvind Neelakantan, Benjamin Roth, Andrew McCallum
Department of Computer Science
University of Massachusetts, Amherst
Amherst, MA, 01003
{arvind,beroth,mccallum}@cs.umass.edu

Abstract

Traditional approaches for knowledge base completion are based on symbolic representations of knowledge. Low-dimensional vector embedding models proposed recently for this task are attractive since they generalize to possibly unlimited sets of relations. A significant drawback of previous embedding models for KB completion is that they merely support reasoning on individual relations (e.g., $bornIn(X, Y) \Rightarrow nationality(X, Y)$). In this work, we develop an embedding model for KB completion that supports *chains of reasoning* on paths of any length using compositional vector space models. Unlike most previous methods, our approach can generalize to paths that are unseen in training and additionally, in a *zero-shot* setting, predict target relations without explicitly training for the target relation types. In a challenging large-scale dataset, our method outperforms a simple classifier method and a method that uses pre-trained vectors by 11% and 7% respectively, and performs competitively with a modified stronger baseline. We also show that the zero-shot model without using any direct supervision achieves impressive results by performing significantly better than a random baseline.

1 Introduction

Knowledge base (KB) construction has been a focus of research in natural language understanding, and large KBs have been created, most notably Freebase (Bollacker et al., 2008), YAGO (Suchanek et al., 2007) and NELL (Carlson et al., 2010). These KBs contain several million facts such as *(Barack Obama, presidentOf, USA)* and *(Brad Pitt, marriedTo, Angelina Jolie)*. However, these KBs are incomplete (Min et al., 2013) and are missing important facts, thus jeopardizing their usefulness in downstream tasks. In this work, we focus on binary relation extraction, i.e., relations with two arguments for KB completion.

Traditional KB completion methods (Mintz et al., 2009; Min et al., 2013; Lao et al., 2011; Lao et al., 2012) use symbolic representations of knowledge and are bound to a fixed and hand-built schema that are usually brittle and incomplete. Low-dimensional vector embedding models proposed recently (Riedel et al., 2013; Bordes et al., 2013) are attractive since they generalize to possibly unlimited set of relations. A drawback of previous work in using embedding models for KB completion is that they merely support simple reasoning of the form $A \Rightarrow B$ (e.g., $bornIn(X, Y) \Rightarrow nationality(X, Y)$).

A more general approach for KB completion is to infer missing relation facts of entity pairs using paths (of length greater than or equal to one) connecting them in the KB graph (Schoenmackers et al., 2010; Lao et al., 2011). Here, the KB graph is constructed with the entities as nodes and (typed) edges indicating relations between them. For example, if the KB contains the facts *IsBasedIn(Microsoft, Seattle)*, *StateLocatedIn(Seattle, Washington)* and *CountryLocatedIn(Washington, USA)*, we can infer the fact *CountryOfHeadquarters(Microsoft, USA)* using the rule:

CountryOfHeadquarters(X, Y) $\Leftrightarrow$
IsBasedIn(X,A) $\wedge$ *StateLocatedIn(A, B)* $\wedge$ *CountryLocatedIn(B, Y)*

where *IsBasedIn - StateLocatedIn - CountryLocatedIn* is a path connecting the entity pair *(Microsoft, USA)* in the KB graph, and *IsBasedIn, StateLocatedIn* and *CountryLocatedIn* are the binary relations in the path.

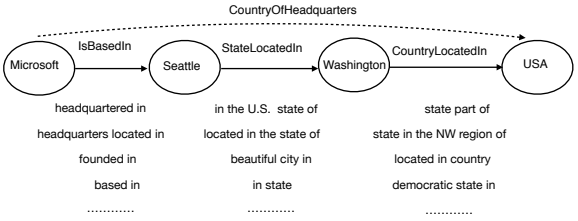

Figure 1: Semantically similar paths connecting entity pair (Microsoft, USA).

A drawback of most previous work (Schoenmackers et al., 2010; Lao et al., 2011; Lao et al., 2012) that uses symbolic representations of knowledge is that they reason on individual paths in the KB graph independently. For example, Lao et al. (2011) create a separate feature for each of the semantically similar paths in Figure 1 leading to feature space explosion and poor generalization. This limits the applicability of these methods to modern KBs that have thousands of relations since the number of paths increases rapidly with the number of relations. Moreover, it is often beneficial to add more information in the form of Subject-Verb-Object (SVO) triples to the KB graph which dramatically increases the number of relations and paths in the KB graph, making the feature explosion problem more severe.

In response, Gardner et al. (2013) and Gardner et al. (2014) use pre-trained low-dimensional vector representations of relations to alleviate the feature explosion problem. Gardner et al. (2013) replace relation type with their cluster membership which reduces the number of distinct paths in the KB graph but the clustering of relations does not capture asymmetric implicature and could lead to loss of important information. Gardner et al. (2014) transform new unseen paths to seen paths by replacing relation types in the unseen paths with relations that are *close* to it in the vector space. Both these methods aim to obtain higher quality paths connecting the entity pairs but still perform inference in the symbolic space and in the path finding step use pre-trained relation vectors that are not tailored for the task. Unlike our model, they do not have the ability to make predictions about relation types that are absent during training (zero-shot learning) .

Our approach performs inference directly in the vector space by comparing the vector representation of a path with the vector representation of the relation to be predicted. We construct compositional vector representations for the paths in the KB graph from the semantic vector representations of the binary relations in that path (Figure 2). We use Recursive Neural Networks (RNNs) (Socher et al., 2011) to model semantic composition. The reasons for using composition models for this task is motivated by : (1) unlike classifiers, they allow us to share parameters across semantically similar paths using the vector representations of the relations in those paths and (2) at test time, we can perform inference using paths that are unseen during training. These advantages empower our model to seamlessly perform inference on millions of paths in the KB graph. Additionally, by learning a single powerful composition function over the semantic vector space and fixing the relation vectors using pre-trained vectors from Riedel et al. (2013), our method can perform zero-shot inference to predict relational facts without explicitly training for the target (or test) relation types.

We evaluate our methods on a large-scale dataset constructed with Freebase (Bollacker et al., 2008) as the KB enriched with entity linked text triples from Clueweb (Orr et al., 2013). This dataset has the following advantages over the dataset used in Gardner et al. (2014): (1) the methods are evaluated to perform inference on an average of over 2 million paths per relation type compared to 1000 in the previous dataset, (2) we use on an average more than $10,000$ entity pairs per relation type from Freebase for training and testing instead of just 200 and (3) while the previous dataset was created by representing each textual entity mention with a separate node in the graph we create nodes only for Freebase entities leveraging the publicly available entity linked information (Orr et al., 2013). We make this dataset containing millions of paths per relation type publicly available.

In this challenging large-scale dataset, our method outperforms the simple classifier method of Lao et al. (2012) and the method of Gardner et al. (2013) that uses pre-trained vectors by 11% and 7% respectively and performs competitively with a modified stronger baseline when evaluated on 46 relation types. The best results are obtained by combining the predictions of our model with the predictions of the modified baseline. This combination achieves 15% and 19% improvement over the method of Lao et al. (2012) and Gardner et al. (2013) respectively. We also show that the

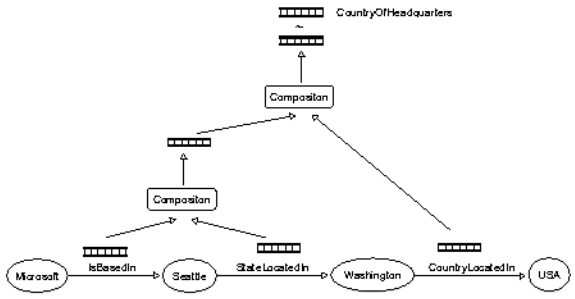

Figure 2: Vector Representations of the paths are computed by applying the composition function recursively.

zero-shot model without explicitly training for the target relation types achieves impressive results by performing significantly better than a random baseline.

2 Background

In this section, we give background on the Path Ranking Algorithm (PRA) which is used to obtain a set of paths connecting an entity pair in a large KB graph and the RNN model which we employ to model the composition function.

2.1 Path Ranking Algorithm

Since it is impractical to exhaustively obtain the set of all paths connecting an entity pair in the large KB graph, we use PRA (Lao et al., 2011) to obtain a set of paths connecting the entity pairs. Given a training set of entity pairs for a relation, PRA heuristically finds a set of paths by performing random walks from the source and target nodes keeping the most common paths. We use PRA to find millions of distinct paths per relation type. We do not use the random walk probabilities given by PRA since using it did not yield improvements in our experiments.

2.2 Recursive Neural Network

RNN model (Socher et al., 2011) constructs vector representation for phrases or sentences (of any length) in natural language by recursively applying a composition function along the parse tree of a sentence. For example, the vector representation of a parent phrase in the parse tree of a sentence consisting of children w_1 and w_2 is given by $f(W[v(w_1); v(w_2)])$ where $v(w) \in \mathbb{R}^d$ is the vector representation of w, f is an element-wise non linearity function, $[a; b]$ represents the concatenation two vectors a and b along with a bias term, and $W \in \mathbb{R}^{d \times 2*d+1}$ is the composition matrix. The model in Socher et al. (2011) is trained with a sentence-level objective using backpropagation through structure (Goller and Küchler, 1996). We make an important modification to the training procedure which is described in the next section making it more suitable for our task.

3 Recursive Neural Networks for KB Inference

In this section, we propose a RNN model for KB inference. The RNN model handles the feature space explosion problem by sharing parameters between semantically similar paths using the vector representations of the relations in those paths. Moreover, they can perform inference at test time using paths that are unseen during training. This characteristic of the model is highly desirable in current KBs since they have thousands of relations and the number of paths in the KB graph increases rapidly with the number of relations.

We represent each binary relation using a d-dimensional real valued vector. Inference is performed by comparing the vector representation of the path with the vector representation of the relation to be predicted using the sigmoid function. The vector representations of the paths (of any length) in the KB graph are computed by applying the composition function recursively as shown in Figure 2. We model composition using the method in Socher et al. (2011). The composition function computes a d-dimensional real valued vector for higher nodes in the tree using the vector representation of the node's two children nodes. We learn a separate composition matrix for every relation that is predicted.

Let $v_r(\delta) \in \mathbb{R}^d$ be the vector representation of relation δ and $v_p(\pi) \in \mathbb{R}^d$ be the vector representation of path π. $v_p(\pi)$ denotes the relation vector if path π is of length one. To predict relation $\delta = \textit{CountryOfHeadquarters}$, the vector representation of the path $\pi = \textit{IsBasedIn - StateLocatedIn}$ containing two relations *IsBasedIn* and *StateLocatedIn* is computed by (Figure 2),

$$v_p(\pi) = f(W_\delta[v_r(\textit{IsBasedIn}); v_r(\textit{StateLocatedIn})])$$

where $f = sigmoid$ is the element-wise nonlinearity function, $W_\delta \in \mathbb{R}^{d*2d+1}$ is the composition matrix for $\delta = \textit{CountryOfHeadquarters}$ and $[a; b]$ represents the concatenation of two vectors

Algorithm 1 Training Algorithm of RNN model for relation δ

Input: $\Lambda_\delta = \Lambda_\delta^+ \cup \Lambda_\delta^-$, Φ_δ, number of iterations T, mini-batch size B
Initialize v_r, W_δ randomly
for $t = 1, 2, \ldots, T$ **do**
    $\nabla v_r = 0$, $\nabla W_\delta = 0$ and $b = 0$
    for $\lambda = (\gamma, \delta) \in \Lambda_\delta$ **do**
        $\mu_\lambda = \arg\max_{\pi \in \Phi_\delta(\gamma)} v_p(\pi).v_r(\delta)$
        Accumulate gradients to ∇v_r, ∇W_δ using path μ_λ.
        $b = b + 1$
        if $b = B$ **then**
            Gradient Update for v_r, W_δ
            $\nabla v_r = 0$, $\nabla W_\delta = 0$ and $b = 0$
        end if
    end for
    if $b > 0$ **then**
        Gradient Update for v_r, W_δ
    end if
end for
Output: v_r, W_δ

$a \in \mathbb{R}^d, b \in \mathbb{R}^d$ along with a bias feature to get a new vector $[a; b] \in \mathbb{R}^{2d+1}$.

The vector representation of the path Π = *IsBasedIn - StateLocatedIn - CountryLocatedIn* in Figure 2 is computed similarly by,

$$v_p(\Pi) = f(W_\delta[v_p(\pi); v_r(\textit{ CountryLocatedIn})])$$

where $v_p(\pi)$ is the vector representation of path *IsBasedIn - StateLocatedIn*. While computing the vector representation of a path we always traverse left to right, composing the relation vector in the right with the accumulated path vector in the left[1]. This makes our model similar to a recurrent neural network (Werbos, 1990).

Finally, we make a prediction regarding *CountryOfHeadquarters(Microsoft, USA)* using the path Π = *IsBasedIn - StateLocatedIn - CountryLocatedIn* by comparing the vector representation of the path ($v_p(\Pi)$) with the vector representation of the relation *CountryOfHeadquarters* (v_r(*CountryOfHeadquarters*)) using the sigmoid function.

3.1 Model Training

We assume that we are given a KB (for example, Freebase enriched with SVO triples) containing a set of entity pairs Γ, set of relations Δ and a set of observed facts Λ^+ where $\forall\lambda = (\gamma, \delta) \in \Lambda^+ (\gamma \in \Gamma, \delta \in \Delta)$ indicates a positive fact that entity pair γ is in relation δ. Let $\Phi_\delta(\gamma)$ denote the set of paths connecting entity pair γ given by PRA for predicting relation δ.

In our task, we only observe the set of paths connecting an entity pair but the path(s) that is predictive of the fact is unobserved. We treat this as a latent variable (μ_λ for the fact λ) and we assign μ_λ the path whose vector representation has maximum dot product with the vector representation of the relation to be predicted. For example, μ_λ for the fact $\lambda = (\gamma, \delta) \in \Lambda^+$ is given by,

$$\mu_\lambda = \arg\max_{\pi \in \Phi_\delta(\gamma)} v_p(\pi).v_r(\delta)$$

Selecting only the path which is closest to the relation in vector space not only allows for faster training (compared to marginalization) but also gave improved performance. This technique has been successfully used in other models previously (Weston et al., 2013; Neelakantan et al., 2014). During training, we assign μ_λ using the current parameter estimates. We use the same procedure to assign μ_λ for unobserved facts that are used as negative examples during training. Note that this scenario does not occur in previous work that use RNNs (Socher et al., 2011; Socher et al., 2012; Socher et al., 2013b; Iyyer et al., 2014; Irsoy and Cardie, 2014).

We train a separate RNN model for predicting each relation and the parameters of the model for predicting relation $\delta \in \Delta$ are $\Theta = \{v_r(\omega) \forall \omega \in \Delta,\ W_\delta\}$. Given a training set consisting of positive (Λ_δ^+) and negative (Λ_δ^-) instances[2] for relation δ, the parameters are trained to maximize the log likelihood of the training set with L-2 regularization.

$$\Theta^* = \arg\max_\Theta \sum_{\lambda=(\gamma,\delta)\in\Lambda_\delta^+} P(y_\lambda = 1; \Theta) + \sum_{\lambda=(\gamma,\delta)\in\Lambda_\delta^-} P(y_\lambda = 0; \Theta) - \rho\|\Theta\|^2$$

where y_λ is a binary random variable which takes the value 1 if the fact λ is true and 0 otherwise, and the probability of a fact $P(y_\lambda = 1; \Theta)$ is given by,

$$P(y_\lambda = 1; \Theta) = sigmoid(v_p(\mu_\lambda).v_r(\delta))$$
$$where\ \ \mu_\lambda = \arg\max_{\pi \in \Phi_\delta(\gamma)} v_p(\pi).v_r(\delta)$$

[1] we did not get significant improvements when we tried more sophisticated ordering schemes for computing the path representations.

[2] we sub-sample a portion of the set of all unobserved instances.

and $P(y_\lambda = 0; \Theta) = 1 - P(y_\lambda = 1; \Theta)$. The relation vectors and the composition matrix are initialized randomly. We train the network using backpropagation through structure (Goller and Küchler, 1996).

4 Zero-shot KB Inference

In this section, we show that our model described in the previous section is capable of zero-shot or zero-data learning after making a few simple modifications. In zero-shot or zero-data learning (Larochelle et al., 2008; Palatucci et al., 2009), few labels or classes is omitted during training the model and only a description of those classes are given at prediction time. We make two modifications to the model described in the previous section, (1) learn a general composition matrix, and (2) fix relation vectors with pre-trained vectors, so that we can predict relations that are unseen during training.

We learn a general composition matrix for all relations instead of learning a separate composition matrix for every relation to be predicted. So, for example, the vector representation of the path π = *IsBasedIn - StateLocatedIn* containing two relations *IsBasedIn* and *StateLocatedIn* is computed by (Figure 2),

$$v_p(\pi) = f(W[v_r(\textit{IsBasedIn}); v_r(\textit{StateLocatedIn})])$$

where $W \in \mathbb{R}^{d*2d+1}$ is the general composition matrix.

The vector representation of the path Π = *IsBasedIn - StateLocatedIn - CountryLocatedIn* in Figure 2 is computed similarly by,

$$v_p(\Pi) = f(W[v_p(\pi); v_r(\textit{CountryLocatedIn})])$$

where $v_p(\pi)$ is the vector representation of path *IsBasedIn - StateLocatedIn*.

We initialize the vector representations of the binary relations (v_r) using the representations learned in Riedel et al. (2013) and do not update them during training. The relation vectors are not updated because at prediction time we would be predicting relation types which are never seen during training and hence their vectors would never get updated. We learn only the general composition matrix in this model. The parameters of the composition matrix are learned using the available training data containing instances of few relations. The other aspects of the model remain unchanged and training is again done using backpropagation through structure (Goller and Küchler, 1996).

To predict facts whose relation types are unseen during training, we compute the vector representation of the path using the general composition matrix and compute the probability of the fact using the pre-trained relation vector. For example, using the vector representation of the path Π = *IsBasedIn - StateLocatedIn - CountryLocatedIn* ($v_p(\Pi)$) in Figure 2, we can predict any relation irrespective of whether they are seen at training by comparing it with the pre-trained relation vectors (v_r). This ability of the model to generalize to unseen relations is beyond the capabilities of all previous methods for KB inference (Schoenmackers et al., 2010; Lao et al., 2011; Gardner et al., 2013; Gardner et al., 2014).

5 Related Work

KB Inference: Methods such as Lin and Pantel (2001), Yates and Etzioni (2007) and Berant et al. (2011) learn inference rules of length one. Schoenmackers et al. (2010) learn general inference rules by considering the set of all paths in the KB and selecting paths that satisfy a certain precision threshold. Their method does not scale well to modern KBs and also depends on carefully tuned thresholds. Lao et al. (2011) train a simple logistic regression classifier with NELL KB paths as features to perform KB completion while Gardner et al. (2013) and Gardner et al. (2014) extend it by using pre-trained relation vectors to overcome feature sparsity. Yang et al. (2014) learn inference rules using simple element-wise addition or multiplication as the composition function.

Compositional Vector Space Models: There has been plenty of work on developing compositional vector space models to represent phrases and sentences in natural language as vectors (Mitchell and Lapata, 2008; Baroni and Zamparelli, 2010; Yessenalina and Cardie, 2011). RNNs (Socher et al., 2011) have been successfully used to learn vector representations of phrases using the vector representations of the words in that phrase. They have been used for many tasks like parsing (Socher et al., 2011), sentiment classification (Socher et al., 2012; Socher et al., 2013b; Irsoy and Cardie, 2014), question answering (Iyyer et al., 2014) and natural language logical semantics

Entities	18M
Freebase Facts	40M
Clueweb triples	12M
Relations	25,994
Relation types tested	46
Avg. paths/relation	2.3M
Avg. training facts/relation	6638
Avg. positive test instances/relation	3492
Avg. negative test instances/relation	43,160

Table 1: Statistics of our dataset.

(Bowman et al., 2014). Recurrent neural networks have been used for many tasks such as language modeling (Mikolov et al., 2010), machine translation (Sutskever et al., 2014) and parsing (Vinyals et al., 2014).

Zero-shot or zero-data learning: Zero-data learning was introduced in Larochelle et al. (2008) for character recognition and drug discovery. Palatucci et al. (2009) perform zero-shot learning for neural decoding while there has been plenty of work in this direction for image recognition (Socher et al., 2013a; Frome et al., 2013; Norouzi et al., 2014).

6 Experiments

In this section, we give details on our data and results. All the neural network models are trained for 150 iterations using 50 dimensional relation vectors, and we set the L2-regularizer and learning rate to 0.0001 and 0.1 respectively. We halved the learning rate after every 60 iterations and use mini-batches of size 20. The neural networks and the classifiers were optimized using AdaGrad (Duchi et al., 2011). The hyperparameters of both the models were tuned on the same development data.

6.1 Data

We ran experiments on Freebase (Bollacker et al., 2008) knowledge base enriched with information from Clueweb. We use the publicly available entity links (Orr et al., 2013) to Freebase in the Clueweb dataset. Hence, we create nodes only for Freebase entities in our KB graph. We remove facts containing /type/object/type as they do not give useful predictive information for our task. We get triples from Clueweb by considering sentences that contain two entities linked to Freebase. We extract the phrase between the two entities and treat them as the relation types. For phrases that are of length greater than four we keep only the first and last two words. This helps us to avoid the time consuming step of dependency parsing the sentence to get the relation type. These triples are similar to facts obtained by OpenIE (Banko et al., 2007). To reduce noise, we select relation types that occur at least 50 times. We evaluate on 46 relation types in Freebase that have the most number of instances. The methods are evaluated on a subset of facts in Freebase that were hidden during training. Table 1 shows important statistics of our dataset.

6.2 Predictive Paths

Table 2 shows predictive paths for 4 relations learned by the RNN model. The high quality of unseen paths is indicative of the fact that the RNN model is able to generalize to paths that are never seen during training.

6.3 Results

We compare the performance of the following methods in our experiments:

PRA Classifier is the method in Lao et al. (2012) which trains a logistic regression classifier by creating a feature for every path type.

Cluster PRA Classifier is the method in Gardner et al. (2013) which replaces relation types from Clueweb triples with their cluster membership in the KB graph before the path finding step. After this step, their method proceeds in exactly the same manner as Lao et al. (2012) training a logistic regression classifier by creating a feature for every path type. We use pre-trained relation vectors from Riedel et al. (2013) and use k-means clustering to cluster the relation types to 25 clusters as done in Gardner et al. (2013).

Composition-Add uses a simple element-wise addition followed by sigmoid non-linearity as the composition function (Yang et al., 2014).

RNN-random is the supervised RNN model described in section 3 with the relation vectors initialized randomly.

RNN is the supervised RNN model described in section 3 with the relation vectors initialized using the method in Riedel et al. (2013).

PRA Classifier-b is our simple extension to the method in Lao et al. (2012) which additionally uses bigrams in the path as features. We add a special *start* and *stop* symbol to the path before computing the bigram features.

Cluster PRA Classifier-b is our simple extension to the method in Gardner et al. (2013) which ad-

Relation: /book/written_work/original_language/ *(book "x" written in language "y")* **Seen paths:** /book/written_work/previous_in_series - /book/written_work/author-/people/person/nationality - /people/person/nationality^{-1} - /people/person/languages /book/written_work/author - /people/ethnicity/people^{-1} - /people/ethnicity/languages_spoken **Unseen paths:** "in"$^{-1}$ - "writer"$^{-1}$ - /people/person/nationality^{-1} - /people/person/languages /book/written_work/author - addresses - /people/person/nationality^{-1} - /people/person/languages
Relation: /people/person/place_of_birth/ *(person "x" born in place "y")* **Seen paths:** "was,born,in" - /location/mailing_address/citytown^{-1} - /location/mailing_address/state_province_region "from" - /location/location/contains^{-1} **Unseen paths:** "born,in" - /location/location/contains - "near"$^{-1}$ "was,born,in" - commonly,known,as^{-1}
Relation: /geography/river/cities/ *(river "x" flows through or borders "y")* **Seen paths:** "at" - /location/location/contains^{-1} "meets,the" - /transportation/bridge/body_of_water_spanned^{-1} - /location/location/contains^{-1} - "in" **Unseen paths:** /geography/lake/outflow^{-1} - /location/location/contains^{-1} /geography/lake/outflow^{-1} - /location/location/contains^{-1} - "near"
Relation: /people/family/members/ *(person "y" part of family "x")* **Seen paths:** /royalty/monarch/royal_line^{-1} - /people/person/children - /royalty/monarch/royal_line - /royalty/royal_line/monarchs_from_this_line /royalty/royal_line/monarchs_from_this_line - /people/person/parents^{-1} - /people/person/parents^{-1} - /people/person/parents^{-1} **Unseen paths:** /royalty/monarch/royal_line^{-1} - "leader"$^{-1}$ - "king" - "was,married,to"$^{-1}$ "of,the"$^{-1}$ - "but,also,of" - "married" - "defended"$^{-1}$

Table 2: Predictive paths, accroding to the *RNN* model, for 4 target relations. Two examples of seen and unseen paths are shown for each target relation. Inverse relations are marked by $^{-1}$, i.e, $r(x, y) \implies r^{-1}(y, x), \forall(x, y) \in r$. Relations within quotes are OpenIE (textual) relation types.

ditionally uses bigram features computed as previously described.

RNN + PRA Classifier combines the predictions of *RNN* and *PRA Classifier*. We combine the predictions by assigning the score of a fact as the sum of their rank in the two models after sorting them in ascending order.

RNN + PRA Classifier-b combines the predictions of *RNN* and *PRA Classifier-b* using the technique described previously.

RNN-ensemble is obtained by combining the predictions of five different RNNs. Apart from *RNN* and *RNN-random*, we trained three more RNNs with different random initialization and the performance of the three RNNs individually are 57.09, 57.11 and 56.91. Combining the predictions of the ensemble with *PRA Classifier-b* did not yield improvements over *RNN + PRA Classifier-b*.

Table 3 shows the results of our experiments. We are not able to include the method described in Gardner et al. (2014) since the publicly available implementation does not scale to our large dataset[3]. First, we show that it is better to train the models using all the path types instead of using only the top 1,000 path types as done in previous work (Gardner et al., 2013; Gardner et al., 2014). We can see that the RNN model performs significantly better than the baseline methods of Lao et al. (2012) and Gardner et al. (2013), and performs competitively with classifiers which additionally use bigram features. The performance of the RNN model is not affected by initialization since using random vectors and pre-trained vectors results in similar performance. The bigram features help the classifiers to handle feature sparsity. The best results are obtained by combining the predictions of our model with the classifiers using bigram features.

6.3.1 Zero-shot

Table 4 shows the results of the zero-shot model described in section 4 compared with the fully supervised RNN model (section 3) and a baseline that produces a random ordering of the test facts. We evaluate on randomly selected 10 (out of 46) relation types, hence for the fully supervised version we train 10 RNNs, one for each relation type. For evaluating the zero-shot model, we randomly

[3]The method in Gardner et al. (2014) has a hyperparameter that controls the probability of restarting a random walk and we suspect the run time of their method is very sensitive to it.

	train with top 1000 paths	train with all paths
Method	MAP	MAP
PRA Classifier	43.46	51.31
Cluster PRA Classifier	46.26	53.23
Composition-Add	40.23	45.37
RNN-random	45.52	56.91
RNN	46.61	56.95
PRA Classifier-b	48.09	58.13
Cluster PRA Classifier-b	48.72	58.02
RNN + PRA Classifier	49.92	58.42
RNN + PRA Classifier-b	51.94	**61.17**
RNN-ensemble	-	59.16

Table 3: Results comparing different methods on 46 types. All the methods perform better when trained using all the paths than training using the top $1,000$ paths. When training with all the paths, *RNN* performs significantly ($\rho < 0.005$) better than *PRA Classifier* and *Cluster PRA Classifier*, and competitively with *PRA Classifier-b* and *Cluster PRA Classifier-b*. The best results are obtained by combining the predictions of *RNN* with *PRA Classifier-b* which performs significantly ($\rho < 10^{-5}$) better than both *PRA Classifier-b* and *Cluster PRA Classifier-b*.

split the relations into two sets of equal size and train a zero-shot model on one set and test on the other set. So, in this case we have two RNNs making predictions on relation types that they have never seen during training. As expected, the fully supervised RNN outperforms the zero-shot model by a large margin but the zero-shot model without using any direct supervision clearly performs much better than a random baseline. We expect the performance of the zero-shot model to predict facts about unseen relation types to increase as we train it on more relations.

6.3.2 Discussion

We suspect the RNN model does not capture long-range structural dependencies well since the best results are achieved only after combining the predictions of the RNN with a classifier using bigram features. To overcome this drawback, in future work, we plan to explore compositional models that have a longer memory (Hochreiter and Schmidhuber, 1997; Cho et al., 2014; Mikolov et al., 2014). We also plan to include vector representations for the entities and develop models that address the issue of polysemy in verb phrases (Cheng et al., 2014).

	train with top 1000 paths	train with all paths
Method	MAP	MAP
RNN	43.82	50.10
zero-shot	19.28	20.61
Random	7.59	

Table 4: Results comparing the zero-shot model with supervised RNN and a random baseline on 10 types. RNN is the fully supervised model described in section 3 while zero-shot is the model described in section 4. The zero-shot model without explicitly training for the target relation types achieves impressive results by performing significantly better than a random baseline.

7 Conclusion

We develop a compositional vector space model for knowledge base inference that unlike most previous methods can generalize to paths that are unseen in training. In a challenging large-scale dataset, our method outperforms two baseline methods and performs competitively with a modified stronger baseline. The best results are obtained by combining the predictions of our model with the predictions of the modified base-

line which achieves a 15% and 19% improvement over the method in Lao et al. (2012) and Gardner et al. (2013) respectively. We also show that the zero-shot model without explicitly training for the target relation types achieves impressive results by performing significantly better than a random baseline.

Acknowledgments

This work was supported in part by the Center for Intelligent Information Retrieval, in part by DARPA under agreement number FA8750-13-2-0020, in part by an award from Google, and in part by NSF grant #CNS-0958392. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

References

[Banko et al.2007] Michele Banko, Michael J Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *International Joint Conference on Artificial Intelligence.*

[Baroni and Zamparelli2010] Marco Baroni and Roberto Zamparelli. 2010. Nouns are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space. In *Empirical Methods in Natural Language Processing*.

[Berant et al.2011] Jonathan Berant, Ido Dagan, and Jacob Goldberger. 2011. Global learning of typed entailment rules. In *Association for Computational Linguistics*.

[Bollacker et al.2008] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*.

[Bordes et al.2013] Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems.*

[Bowman et al.2014] Samuel R. Bowman, Christopher Potts, and Christopher D Manning. 2014. Recursive neural networks for learning logical semantics. In *CoRR*.

[Carlson et al.2010] Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R. Hruschka, and A. 2010. Toward an architecture for never-ending language learning. In *In AAAI*.

[Cheng et al.2014] Cheng, Jianpeng Kartsaklis, and Edward Grefenstette. 2014. Investigating the role of prior disambiguation in deep-learning compositional models of meaning. In *In Learning Semantics workshop NIPS.*

[Cho et al.2014] Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder–decoder approaches. In *Workshop on Syntax, Semantics and Structure in Statistical Translation*.

[Duchi et al.2011] John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. In *Journal of Machine Learning Research.*

[Frome et al.2013] Andrea Frome, Gregory S. Corrado, Jonathon Shlens, Samy Bengio, Jeffrey Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. 2013. Devise: A deep visual-semantic embedding model. In *Neural Information Processing Systems.*

[Gardner et al.2013] Matt Gardner, Partha Pratim Talukdar, Bryan Kisiel, and Tom M. Mitchell. 2013. Improving learning and inference in a large knowledge-base using latent syntactic cues. In *Empirical Methods in Natural Language Processing*.

[Gardner et al.2014] Matt Gardner, Partha Talukdar, Jayant Krishnamurthy, and Tom Mitchell. 2014. Incorporating vector space similarity in random walk inference over knowledge bases. In *Empirical Methods in Natural Language Processing*.

[Goller and Küchler1996] Christoph Goller and Andreas Küchler. 1996. Learning task-dependent distributed representations by backpropagation through structure. In *IEEE Transactions on Neural Networks.*

[Hochreiter and Schmidhuber1997] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. In *Neural Computation.*

[Irsoy and Cardie2014] Ozan Irsoy and Claire Cardie. 2014. Deep recursive neural networks for compositionality in language. In *Neural Information Processing Systems.*

[Iyyer et al.2014] Mohit Iyyer, Jordan Boyd-Graber, Leonardo Claudino, Richard Socher, and Hal Daumé III. 2014. A neural network for factoid question answering over paragraphs. In *Empirical Methods in Natural Language Processing.*

[Lao et al.2011] Ni Lao, Tom Mitchell, and William W. Cohen. 2011. Random walk inference and learning in a large scale knowledge base. In *Conference on Empirical Methods in Natural Language Processing*.

[Lao et al.2012] Ni Lao, Amarnag Subramanya, Fernando Pereira, and William W. Cohen. 2012. Reading the web with learned syntactic-semantic inference rules. In *Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*.

[Larochelle et al.2008] Hugo Larochelle, Dumitru Erhan, and Yoshua Bengio. 2008. Zero-data learning of new tasks. In *National Conference on Artificial Intelligence.*

[Lin and Pantel2001] Dekang Lin and Patrick Pantel. 2001. Dirt - discovery of inference rules from text. In *International Conference on Knowledge Discovery and Data Mining*.

[Mikolov et al.2010] Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernocký, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Annual Conference of the International Speech Communication Association*.

[Mikolov et al.2014] Tomas Mikolov, Armand Joulin, Sumit Chopra, Michaël Mathieu, and Marc'Aurelio Ranzato. 2014. Learning longer memory in recurrent neural networks. In *CoRR*.

[Min et al.2013] Bonan Min, Ralph Grishman, Li Wan, Chang Wang, and David Gondek. 2013. Distant supervision for relation extraction with an incomplete knowledge base. In *HLT-NAACL*, pages 777–782.

[Mintz et al.2009] Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Association for Computational Linguistics and International Joint Conference on Natural Language Processing*.

[Mitchell and Lapata2008] Jeff Mitchell and Mirella Lapata. 2008. Vector-based models of semantic composition. In *Association for Computational Linguistics*.

[Neelakantan et al.2014] Arvind Neelakantan, Jeevan Shankar, Alexandre Passos, and Andrew McCallum. 2014. Efficient non-parametric estimation of multiple embeddings per word in vector space. In *Empirical Methods in Natural Language Processing.*

[Norouzi et al.2014] Mohammad Norouzi, Tomas Mikolov, Samy Bengio, Yoram Singer, Jonathon Shlens, Andrea Frome, Greg Corrado, and Jeffrey Dean. 2014. Zero-shot learning by convex combination of semantic embeddings. In *International Conference on Learning Representations.*

[Orr et al.2013] Dave Orr, Amarnag Subramanya, Evgeniy Gabrilovich, and Michael Ringgaard. 2013. 11 billion clues in 800 million documents: A web research corpus annotated with freebase concepts. http://googleresearch.blogspot.com/2013/07/11-billion-clues-in-800-million.html.

[Palatucci et al.2009] Mark Palatucci, Dean Pomerleau, Geoffrey Hinton, and Tom Mitchell. 2009. Zero-shot learning with semantic output codes. In *Neural Information Processing Systems.*

[Riedel et al.2013] Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M. Marlin. 2013. Relation extraction with matrix factorization and universal schemas. In *HLT-NAACL.*

[Schoenmackers et al.2010] Stefan Schoenmackers, Oren Etzioni, Daniel S. Weld, and Jesse Davis. 2010. Learning first-order horn clauses from web text. In *Empirical Methods in Natural Language Processing*.

[Socher et al.2011] Richard Socher, Cliff Chiung-Yu Lin, Christopher D. Manning, and Andrew Y. Ng. 2011. Parsing natural scenes and natural language with recursive neural networks. In *Proceedings of the 26th International Conference on Machine Learning (ICML)*.

[Socher et al.2012] Richard Socher, Brody Huval, Christopher D. Manning, and Andrew Y. Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*.

[Socher et al.2013a] Richard Socher, Milind Ganjoo, Christopher D Manning, and Andrew Ng. 2013a. Zero-shot learning through cross-modal transfer. In *Neural Information Processing Systems.*

[Socher et al.2013b] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013b. Recursive deep models for semantic compositionality over a sentiment treebank. In *Conference on Empirical Methods in Natural Language Processing*.

[Suchanek et al.2007] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: A core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web*.

[Sutskever et al.2014] Ilya Sutskever, Oriol Vinyals, and Quoc V. V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems.*

[Vinyals et al.2014] Oriol Vinyals, Lukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. 2014. Grammar as a foreign language. In *CoRR.*

[Werbos1990] Paul Werbos. 1990. Backpropagation through time: what it does and how to do it. In *IEEE.*

[Weston et al.2013] Jason Weston, Ron Weiss, and Hector Yee. 2013. Nonlinear latent factorization by embedding multiple user interests. In *ACM International Conference on Recommender Systems.*

[Yang et al.2014] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2014. Embedding entities and relations for learning and inference in knowledge bases. In *CoRR*.

[Yates and Etzioni2007] Alexander Yates and Oren Etzioni. 2007. Unsupervised resolution of objects and relations on the web. In *North American Chapter of the Association for Computational Linguistics*.

[Yessenalina and Cardie2011] Ainur Yessenalina and Claire Cardie. 2011. Compositional matrix-space models for sentiment analysis. In *Empirical Methods in Natural Language Processing*.