Compositional Vector Space Models for Knowledge Base Inference

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 to knowledge base completion have been based on symbolic representations. 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 models for KB completion that support *chains of reasoning* on paths of any length using compositional vector space models. We construct compositional vector representations for the paths in the KB graph from the semantic vector representations of the binary relations in that path and perform inference directly in the vector space. Unlike previous methods, our approach can generalize to paths that are unseen in training and, in a *zero-shot* setting, predict target relations without supervised training data for that relation.

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, Kasneci, and Weikum 2007) and NELL (Carlson et al. 2010). These KBs contain several million facts such as *(Barack Obama, presidentOf, USA)* and *(Tom Brady, memberof, New England Patriots)*. However, these KBs are incomplete (Min et al. 2013) and are missing important facts, thus jeopardizing their usefulness for downstream tasks.

Previous work in KB completion (Mintz et al. 2009; Lao, Mitchell, and Cohen 2011; Lao et al. 2012) use symbolic representations of knowledge and are bound to a fixed and hand-built schema. 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., $bornIn(X, Y) \Rightarrow nationality(X, Y)$).

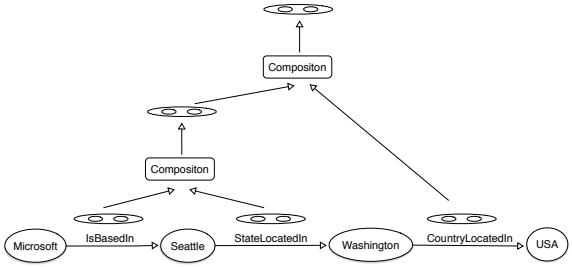

Figure 1: Vector Representations of the paths are computed by applying the composition function recursively.

A more general approach for KB completion is to infer missing relation facts of entity pairs using paths connecting them in the KB graph (Lao, Mitchell, and Cohen 2011; Gardner et al. 2013). 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) :- IsBasedIn(X,A) $\wedge$ *StateLocatedIn(B, C)* $\wedge$ *CountryLocatedIn(C, Y)*

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

Using paths as separate features to predict missing relations (Schoenmackers et al. 2010; Lao, Mitchell, and Cohen 2011; Lao et al. 2012), however, leads to feature space explosion and poor generalization. Modern KBs have thousands of relations and the number of paths is exponential in 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 further increases the number of relations and paths in the KB graph, making the feature explosion problem more severe. In fact, Gardner et al. note that the performance of these methods drop significantly as more facts are added to the KB.

Our approach constructs compositional vector representations for the paths in the KB graph from the semantic vector

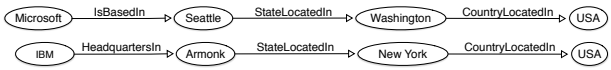

Figure 2: Two semantically similar paths connecting entity pairs (Microsoft, USA) and (IBM, USA).

representations of the binary relations present in that path. We use Recursive Neural Networks (RNNs) (Socher et al. 2011) to model semantic composition. Unlike previous approaches (Schoenmackers et al. 2010; Lao, Mitchell, and Cohen 2011; Lao et al. 2012), our model can predict using paths that are unseen in training, if the relations in the path are observed in the training data. This ability to generalize is crucial in modern KBs that have millions of paths connecting entity pairs in the KB.

Experimental results show that our approach outperforms previous methods when predicting relations having large number of unseen paths and best performance is achieved by combining the predictions of our approach with previous work. We also develop a *zero-shot* model that achieves reasonable performance (well above a random baseline) without using any training data for the relation that it is predicting.

2 Recursive Neural Networks for KB Inference

In our model for KB inference, each binary relation is represented using a d-dimensional real valued vector. The vector representations of the paths in the KB are computed by applying the composition function recursively as shown in Figure 1. The composition function takes two d-dimensional real valued vectors as input and outputs a new d-dimensional real valued vector which can be recursively used to compute vector representations for paths of any length. To model composition, we adopt the method in Socher et al., where the composition operation is performed using a matrix $W_r \in \mathbb{R}^{d*2d}$. Given the vector representation of the two children ($c_1 \in \mathbb{R}^d, c_2 \in \mathbb{R}^d$), the vector representation of the parent $p \in \mathbb{R}^d$ is given by $p = f(W_r[c_1; c_2])$, where $f = tanh$ is the element-wise non-linearity function, $[a; b]$ represents the concatenation of two vectors $a \in \mathbb{R}^d, b \in \mathbb{R}^d$ to get a new vector $[a; b] \in \mathbb{R}^{2d}$ and W_r is a matrix learned to perform composition operation for predicting relation r between the entity pairs.

We predict the missing relations of an entity pair using the vector representations of the path connecting them. To predict missing relations, we compare the learned vector representation of the relation to be predicted with the vector representation of the path constructed using the composition function (Socher et al. 2013).

Compositional vector space models help in handling the feature space explosion problem faced by the classifier approach. For example in Figure 2, two entity pairs *(Microsoft, USA)* and *(IBM, USA)* are connected using semantically similar paths. Methods like Lao, Mitchell, and Cohen and Lao et al. create two different features for these semantically similar paths. So, there is no sharing of parameters between these two paths each having three binary relations, out of which two of them are exactly the same and the other relation pair is semantically equivalent. This leads to data sparsity issues and drop in performance on large KBs (Gardner et al. 2013). Our model overcomes this issue by constructing vector representations for a path using the vector representations of the binary relations in that path.

Modern KBs have thousands of relations and the number of paths in the knowledge graph is exponential in the number of relations. Hence, handling unseen paths is crucial to achieve good performance. By creating a feature for every path in the knowledge graph previous methods at prediction time cannot handle paths that are not observed in the training data. In contrast, our method can construct vector representation of paths that are unseen while training, if the binary relations in the path are observed in the training data.

Pre-Trained Vectors for Binary Relations

We initialize the vector representations of the binary relations using the representations learned in Riedel et al. which is useful for the following reasons: (1) Good initialization could lead to a better local optimum solution since the objective function is non-convex. (2) At test time unlike previous approaches (Lao, Mitchell, and Cohen 2011; Lao et al. 2012), our method can handle any binary relation in the KB even if they are not seen in the training data. By using the pre-trained vector representations of the binary relations we can estimate the vector representation of the path even if the binary relations in that path are never observed during training.

3 Zero-shot KB Inference

In zero-shot or zero-data learning (Larochelle, Erhan, and Bengio 2008), training data for a few classes is omitted and a description of those classes is given only at prediction time. We propose a zero-shot model for KB inference where we can predict relations that are unseen during training. Here, instead of learning a separate composition matrix for every relation that is being predicted we learn a single composition matrix. The composition operation is always performed using a matrix $W \in \mathbb{R}^{d*2d}$ irrespective of the relation to be predicted. Given the vector representation of the two children ($c_1 \in \mathbb{R}^d, c_2 \in \mathbb{R}^d$), the vector representation of the parent $p \in \mathbb{R}^d$ is given by $p = f(W[c_1; c_2])$, where W is a general composition matrix learned to construct the vector representation of a path.

We fix the vectors of the relations in the path and the relations to be predicted with vector representations learned using Riedel et al.. A single composition matrix, irrespective of the relation to be predicted empowers the model to make predictions on relation types for which there are no training data examples.

The objective function minimized by this model is convex since the parameters to be learned are only the composition matrix. Hence, training in this model is guaranteed to converge to the global optimum solution.

Metric	LR	LR-b	RNN-r	RNN	RNN + LR	RNN + LR-b	Random	ZS
Average MAP	0.5625	0.5715	0.5505	0.5699	0.5853	**0.5923**	0.2028	0.3833
Weighted MAP	0.5175	0.5324	0.5185	0.5198	0.5403	**0.5492**	0.1475	0.3248
Average MAP (flipped)	0.4668	0.4642	0.4390	0.4707	0.4892	**0.4965**	0.1366	0.3923
Weighted MAP (flipped)	0.5136	0.5139	0.4806	0.4901	0.5289	**0.5297**	0.1376	0.3767

Table 1: Results on the original and flipped dataset. **LR:** logistic regression. **LR-b:** LR with bigram features. **RNN-r:** recursive neural network initialized with random relation vectors. **RNN:** recursive neural network initialized with pre-trained relation vectors. **RNN+LR/LR-b:** Combination of RNN with LR/LR-b. **Random:** Random baseline. **ZS:** Zero-shot model.

Relation	LR	RNN	UP Ratio
stadiumlocatedincity	0.2694	**0.4029**	0.950
countryhascompanyoffice	0.2127	**0.2545**	0.937
cityliesonriver	0.2634	**0.3998**	0.910
headquarteredin	0.3088	**0.3550**	0.910
companyceo	0.7605	**0.8525**	0.828
citylocatedincountry	**0.4326**	0.2073	0.783
locationlocatedwithinlocation	0.3679	**0.3803**	0.710
athleteplaysforteam	0.2404	**0.3469**	0.642
athleteplaysinleague	**0.7600**	0.7077	0.597
writerwrotebook	**0.8845**	0.8129	0.567
publicationjournalist	**0.6718**	0.5801	0.566
teamplaysagainstteam	**0.4301**	0.3487	0.211

Table 2: Per relation results on the flipped dataset. **UP Ratio:** ratio of unseen paths in test.

4 Related Work

KB Inference: Methods such as Lin and Pantel, Yates and Etzioni and Berant, Dagan, and Goldberger learn inference rules of length one. Schoenmackers et al. 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, Mitchell, and Cohen trained a simple logistic regression classifier with NELL KB paths as features. Gardner et al. add SVO triples to the KB graph, and cluster them in order to overcome feature sparsity. Our method is not directly comparable with them since their method operates on a different set of clustered features.

Compositional Vector Space Models: There has been plenty of work on compositional vector space semantics of natural language (Mitchell and Lapata 2008; Baroni and Zamparelli 2010; Yessenalina and Cardie 2011). RNNs have been successfully used to learn vector representations of phrases using the vector representations of the words in that phrase (Socher et al. 2012).

5 Experiments

For all our experiments we train the network for 50 iterations using 25 dimensional vectors for the binary relations and set the L2-regularizer and learning rate to 0.00001 and 0.01 respectively. The neural network is trained using the back propagation algorithm as described in Socher et al.. When we learn a separate composition matrix for every relation to be predicted we update both the vector representations and the composition matrix while in the zero-shot experiments we update only the composition matrix.

First, we tested our approach on the dataset described in Gardner et al.. The methods are evaluated on 12 relations. The dataset contains facts from the NELL KB (Carlson et al. 2010) and SVO triples from Clueweb (Orr et al. 2013). This dataset was constructed by aggressive feature pruning so that the classifier can handle the number of features that are created. Given that the number of relations in the KB graph is $218,913$ the number of paths that were considered for predicting each relation is only 750. This makes the number of unseen paths in prediction time unrealistic. The percentage of unseen paths to the total number of paths in test data is just 1.7% in this dataset, resulting in an unrealistically optimistic setting.

To make the evaluation more realistic, we therefore flipped the dataset by training on the test data and evaluating on the training data. The percentage of unseen paths in this case is on average 71.8%. For the zero-shot model to make predictions on a relation type we train on the examples of the other eleven relations. The number of entity pairs per relation that are used for training averages to 670 for the original dataset and to 77 for the flipped dataset.

The results for both settings are shown in Table 1. Initialization with pre-trained vectors helps the RNN model to achieve better performance. The logistic regression classifier and the RNN give similar performance, while their combination gives best performance on both datasets. Even though the Zero-shot method does not use any supervised training data for the relation being predicted, it performs better than the random baseline method. Table 2 shows the per-relation results for the flipped data-set, sorted by unseen paths ratio.

The RNN approach significantly outperforms the classifier on the five relations having the highest unseen paths ratio.

6 Conclusion

We developed a compositional vector space model for knowledge base inference that unlike previous methods generalizes to paths which are unseen in training. Empirical results show that our method outperforms previous work on predicting relations that have a high unseen paths ratio, and a combination of our model with a classifier based on path features achieves best performance. The zero-shot model can successfully predict missing instances of relation types that are unoberved in training.

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

Baroni, M., and Zamparelli, R. 2010. Nouns are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space. In *Empirical Methods in Natural Language Processing*.

Berant, J.; Dagan, I.; and Goldberger, J. 2011. Global learning of typed entailment rules. In *Association for Computational Linguistics*.

Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T.; and Taylor, J. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*.

Bordes, A.; Usunier, N.; García-Durán, A.; Weston, J.; and Yakhnenko, O. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems.*

Carlson, A.; Betteridge, J.; Kisiel, B.; Settles, B.; Hruschka, E. R.; and A. 2010. Toward an architecture for never-ending language learning. In *In AAAI*.

Gardner, M.; Talukdar, P. P.; Kisiel, B.; and Mitchell, T. M. 2013. Improving learning and inference in a large knowledge-base using latent syntactic cues. In *Empirical Methods in Natural Language Processing*.

Lao, N.; Subramanya, A.; Pereira, F.; and Cohen, W. W. 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*.

Lao, N.; Mitchell, T.; and Cohen, W. W. 2011. Random walk inference and learning in a large scale knowledge base. In *Conference on Empirical Methods in Natural Language Processing*.

Larochelle, H.; Erhan, D.; and Bengio, Y. 2008. Zero-data learning of new tasks. In *National Conference on Artificial Intelligence.*

Lin, D., and Pantel, P. 2001. Dirt - discovery of inference rules from text. In *International Conference on Knowledge Discovery and Data Mining*.

Min, B.; Grishman, R.; Wan, L.; Wang, C.; and Gondek, D. 2013. Distant supervision for relation extraction with an incomplete knowledge base. In *HLT-NAACL*, 777–782.

Mintz, M.; Bills, S.; Snow, R.; and Jurafsky, D. 2009. Distant supervision for relation extraction without labeled data. In *Association for Computational Linguistics and International Joint Conference on Natural Language Processing*.

Mitchell, J., and Lapata, M. 2008. Vector-based models of semantic composition. In *Association for Computational Linguistics*.

Orr, D.; Subramanya, A.; Gabrilovich, E.; and Ringgaard, M. 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.

Riedel, S.; Yao, L.; McCallum, A.; and Marlin, B. M. 2013. Relation extraction with matrix factorization and universal schemas. In *HLT-NAACL.*

Schoenmackers, S.; Etzioni, O.; Weld, D. S.; and Davis, J. 2010. Learning first-order horn clauses from web text. In *Empirical Methods in Natural Language Processing*.

Socher, R.; Lin, C. C.-Y.; Manning, C. D.; and Ng, A. Y. 2011. Parsing natural scenes and natural language with recursive neural networks. In *Proceedings of the 26th International Conference on Machine Learning (ICML)*.

Socher, R.; Huval, B.; Manning, C. D.; and Ng, A. Y. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*.

Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C. D.; Ng, A. Y.; and Potts, C. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Conference on Empirical Methods in Natural Language Processing*.

Suchanek, F. M.; Kasneci, G.; and Weikum, G. 2007. Yago: A core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web*.

Yates, A., and Etzioni, O. 2007. Unsupervised resolution of objects and relations on the web. In *North American Chapter of the Association for Computational Linguistics*.

Yessenalina, A., and Cardie, C. 2011. Compositional matrix-space models for sentiment analysis. In *Empirical Methods in Natural Language Processing*.