Inference over Knowledge Bases using Neural Models

Anshul Bawa
IIT Delhi

Abstract

Within inference models over knowledge bases, neural embedding-based models are mostly used for learning representations of relations and entities as vectors. They are also useful for composing these representations to reason over paths in a knowledge graph. This paper gives an overview of the various approaches to KB inference that involve embedding-based models, with a focus on path-based inference.

1 Introduction

There are many large-scale Knowledge Bases (KBs) like NELL, YAGO, FreeBase, and DBpedia among others, that contain facts in the form of triples and are organized as graphs of entities and the relations between them. Inference over KBs is the task of leveraging existing facts in a KB to infer new facts. These inferred facts can be used to make the KB more complete in terms of coverage, or to facilitate complex reasoning over KBs, like answering path queries or discovering inference rules.

Earlier approaches to reasoning over KBs were symbolic, and learned general rules or Horn clauses over multiple relations to predict of new relations, like in SHERLOCK (Schoenmackers et al., 2010). This model explores all relation paths of increasing lengths, and selects paths with high predictive power. Such an exhaustive exploration is not possible for modern KBs. This is because the number of potential features to learn explodes with the number of distinct paths and relation types, and also because it depends on precision thresholds that need fine-tuning.

This led to the popularization of latent feature models that learned vectors for relations and entities. These latent feature vectors or embeddings are learned by modeling relations as transformations over entities. These vector-to-vector transformations could be translations (Bordes et al., 2013), matrix multiplication (Yang et al., 2014) or more generally non-linear transformations like tensor operations(Socher et al., 2013). Relation embeddings can also be learned based on matrix factorization (Nickel 2011) or Bayesian clustering (Sutskever et al., 2009). These embedding-based models, once trained, have simple computations at prediction time, making them a scalable alternative representation of entities and relations. An advantage of this representation is that novel relations can be predicted over given pairs of entities through simple vector operations. Vector similarity has a neat interpretation as semantic similarity. When the dimensionality of this vector space is kept low, the models learn to generalize to new facts.

The models mentioned so far perform non-path-based relation prediction. More recent work learns embeddings not just from direct relations but also from multi-step relation paths, as it is found that using embeddings trained only on single relations to answer compositional queries often lead to cascading errors (Guu et al., 2015). Composing representations of multiple relations in a path effectively is a challenge because number of possible relation paths grows exponentially with path length. Relation composition for path-based inference has been modeled and learned in various ways : ranging from addition or multiplication of relation vectors (Yang et al., 2014),(Garcia-Duran et al., 2015),(Lin et al., 2015), interpreting paths as a recursive application of translation and membership op-

erators (Guu et al., 2015), to merging consecutive relation vectors using recurrent neural networks (Neelakantan et al., 2015). The latter reasons about path compositions non-atomically, arriving at vector representations for relation paths. Further improvements to these models include the ability to model the intermediate entities of a path, and the ability to incorporate evidence from multiple paths for prediction (Das et al., 2016), (Toutanova et al., 2016). We elaborate on this class of models in Section 2.

There is another line of work in which embeddings are exploited to discover synonymous relation paths, either by clustering paths (Gardner et al., 2013) or by incorporating vector space similarity to random walk path-traversal probabilities (Gardner et al., 2014). Synonymous relations are also composed using convolutional neural networks (Toutanova et al., 2015) to increase the statistical power of the composed models. These methods are particularly useful when the knowledge graph is constructed not just from a knowledge base but from a combination of a knowledge base and a text corpus. Relations that are extracted from text and aligned to the knowledge base have much more noise than the more neat relations in a knowledge base, and such compositions particularly benefit textual relations, by allowing them to share parameters. We talk a little more about these models in Section 3.

Kindly refer to Table 1 for a high-level summary of all models surveyed.

2 Composing relations in a path

One of the first models to explicitly capture the compositional semantics of relations in the form of multi-relation embeddings is DISTMULT, also called BILINEAR-DIAG (Yang et al., 2014). It treats relations as bilinear maps over entity vectors, and compositions of relations are then characterized as matrix multiplication, which are then used to learn inference rules. ¡Formulation¿ The bilinear objective is learnt on a neural network with a loss that maximizes the margin between scores of positive and negative examples.

Another model, called RTRANSE (Garcia-Duran et al., 2015) also learns to explicitly model relation compositions as additions of the corresponding vectors. Since it builds on TRANSE (Bordes et al., 2013), it treats relations as translation vectors over entities. RTRANSE performs constrained walks over the knowledge graph to arrive at path compositions which are then added into the knowledge graph, effectively augmenting the training set. This can be seen as additionally constraining the margin-based scoring in TRANSE to also reproduce compositions of relations and not just individual relations.

PTRANSE (Lin et al., 2015) is yet another model built over TRANSE that uses relation paths, not to augment training triples, but to define a scoring function over candidate relations. In addition, it associates a reliability score to relation paths, to handle path ambiguity. The composed paths serve as features weighted by their random walk probabilities and valued as the vector similarity to the relation being scored. They employ three kinds of composition operations for paths, namely vector addition, multiplication and an RNN-based non-linear transformation. Since relations are simply modeled as translations in the underlying vector space (as opposed to tensor operations (Socher et al., 2013) or matrix multiplication (Yang et al., 2014)), composition by addition performs better than the other two approaches.

All three of these models, BILINEAR-DIAG, RTRANSE and PTRANSE, restrict these compositions to a length of two or three, to specific schema types or functional constraints on relations or reliability scores to contain ambiguity.

(Guu et al., 2015) propose a unifying framework for training on a compositional objective function, interpreting a relation (or an edge traversal in a knowledge graph) as a traversal operator followed by a member-

ship operator. These operators are explicitly used to train a max-margin loss over paths of a bounded length as training examples. This encourages the model to answer path queries by producing what they call set vectors, or the set of entities that can be arrived at after a given sequence of edge traversals from a given entity. This framework is applicable to composable vector spaces like in BILINEAR-DIAG and TRANSE (producing the variants BILINEAR-DIAG COMP and TRANSE COMP), but not applicable to other kinds of embeddings learnt via matrix factorization (Riedel et al., 2013) or tensor transformations (Socher et al., 2013). Because this framework can handle paths of a longer length, it enables models to predict paths, and have a better robustness to missing edges. Since training on paths provides a sort of structural regularization over training just on edges, path-augmented training also improves performance on the link prediction task (predicting single relations).

Improving over both (Lin et al., 2015) and (Guu et al., 2015), the ALL-PATHS model (Toutanova et al., 2016) uses a dynamic programming approach to efficiently incorporate evidence from all relation paths of a bounded length, eliminating the need for approximation through random walk sampling (Guu et al., 2015) or reliability pruning (Lin et al., 2015). This approach works by exploiting overlapping subcomponents of distinct paths, without having to store all paths between two entities. ALL-PATHS scores candidate relations on path similarity features just like in (Lin et al., 2015), instead of using paths as compositional regularizers like in (Guu et al., 2015). An important benefit of the ability to incorporate all paths is that information from intermediate nodes/entities of a path can be incorporated in the path representations, achieved here using scalar weights associated with intermediate entities.

Even though all the above methods perform compositions of relations, they still treat relation edges atomically. An alternative formulation is the PATH-RNN model (Neelakantan et al., 2015), which arrives at a composed vector representation of a path of connected relations by merging these relations via an RNN, which takes the embeddings of binary relations in the path as an input. This is similar to the RNN-based composition in PTRANSE, but is not limited to a translational relational space, and can compose paths of arbitrary lengths. Path-RNN performs well in predicting new relations, even those not seen during training. It generalizes by learning a composite matrix over all relations instead of separate matrices for each relation. This however, has the drawback of losing a lot of local information. Unlike the above-mentioned approaches like ALL-PATHS, this model can only use evidence from one path at a time, and also ignores all intermediate entities on the path. These limitations have been addressed in (Das et al., 2016), where they add neural attention models and score pooling to incorporate evidence from multiple paths while making a prediction, and also train jointly over relations, entities and entity types.

3 Composing synonymous relation paths

Augmenting the neural embeddings from a KB with a text corpus to improve inference has been explored in (Gardner et al., 2013) and (Gardner et al., 2014), which attempt to alleviate the sparsity inherent in text-based relations, by clustering synonymous relations together CLUSTERPRA and by incorporating vector space similarity to random walk path-traversal probabilities SIMILARPRA, respectively. The vector representations in (Gardner et al., 2013) and (Gardner et al., 2014) are based on the way the textual relations co-occur.

Another approach to alleviate this sparsity is to share learning among different lexical relations that have a similar meaning, by incorporating not just their co-occurrence pattern but also their compositional structure. This is the approach adopted in (Toutanova et al., 2015). They seek to augment existing models like with a convolutional neural

network that composes embeddings of various textual relations to arrive at an embedding for a full lexicalized dependency path. Dependency paths are a natural way of aligning textual relations to KB entities, following previous work (Riedel et al., 2013). In the combined knowledge graph, each such dependency path is treated as a relation. With the insight that synonymous relations often share a lot of common entities/arcs in their dependency paths, the CNN shares parameters among related dependency paths and improves its statistical strength. So whereas the underlying model like DISTMULT learns distinct feature vectors for all relation types, its convolutional variant CONV-DISTMULT derives feature vectors for relation types as the final layer of a CNN built on shared underlying dependency arcs and entities.

References

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

Rajarshi Das, Arvind Neelakantan, David Belanger, and Andrew McCallum. 2016. Chains of reasoning over entities, relations, and text using recurrent neural networks. *arXiv preprint arXiv:1607.01426* .

Alberto Garcia-Duran, Antoine Bordes, and Nicolas Usunier. 2015. *Composing relationships with translations*. Ph.D. thesis, CNRS, Heudiasyc.

Matt Gardner, Partha Pratim Talukdar, Bryan Kisiel, and Tom Mitchell. 2013. Improving learning and inference in a large knowledge-base using latent syntactic cues .

Matt Gardner, Partha Pratim Talukdar, Jayant Krishnamurthy, and Tom Mitchell. 2014. Incorporating vector space similarity in random walk inference over knowledge bases .

Kelvin Guu, John Miller, and Percy Liang. 2015. Traversing knowledge graphs in vector space. *arXiv preprint arXiv:1506.01094* .

Yankai Lin, Zhiyuan Liu, Huanbo Luan, Maosong Sun, Siwei Rao, and Song Liu. 2015. Modeling relation paths for representation learning of knowledge bases. *arXiv preprint arXiv:1506.00379* .

Arvind Neelakantan, Benjamin Roth, and Andrew McCallum. 2015. Compositional vector space models for knowledge base completion. *arXiv preprint arXiv:1504.06662* .

Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M Marlin. 2013. Relation extraction with matrix factorization and universal schemas .

Stefan Schoenmackers, Oren Etzioni, Daniel S Weld, and Jesse Davis. 2010. Learning first-order horn clauses from web text. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 1088–1098.

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

Ilya Sutskever, Joshua B Tenenbaum, and Ruslan R Salakhutdinov. 2009. Modelling relational data using bayesian clustered tensor factorization. In *Advances in neural information processing systems*. pages 1821–1828.

Kristina Toutanova, Danqi Chen, Patrick Pantel, Hoifung Poon, Pallavi Choudhury, and Michael Gamon. 2015. Representing text for joint embedding of text and knowledge bases. In *EMNLP*. Citeseer, volume 15, pages 1499–1509.

Kristina Toutanova, Xi Victoria Lin, Wen-tau Yih, Hoifung Poon, and Chris Quirk. 2016. Compositional learning of embeddings for relation paths in knowledge bases and text. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. volume 1, pages 1434–1444.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2014. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575* .

Model	Entity	Relation	Relation operation	Composition	Path selection criteria	Testing on paths	Textual relations	Paths as features	Paths as regularization	Relation-specificity	Datasets	Metrics
RTRANSE	Vector	Vector	Vector addition	Vector addition	Constrained random walks	No	No	No	Yes	Same for all	FB15k	MRR, HITS@10
PTRANSE	Vector	Vector	Vector addition	Concatenation	Reliability from flow	No	Yes	Yes	No	Same for all	FB15k, FB40k	MRR, HITS@1
TRANSE COMP	Vector	Vector	Vector addition	Vector addition	Random walks	Yes	No	No	Yes	Same for all	WordNet, FB	MQ, HITS@10
BILINEAR-DIAG COMP	Vector	Bilinear map	Bilinear multiplication	Bilinear multiplication	Random walks	Yes	No	No	Yes	Same for all	WordNet, FB	MQ, HITS@10
ALL-PATHS	Vector	Bilinear map	Bilinear multiplication	Dynamic programming	Path length	No	Yes	Yes	No	Same for all	WordNet	MAP, HITS@10
PATH-RNN	None	Vector	-	RNN	Single path highest cosine similarity	Yes	Yes	No	No	Separate for each	FB + ClueWeb	MAP
PATH-RNN JOINT	Vector	Vector	Vector addition	RNN	Cosine similarity from multiple paths	Yes	Yes	Yes	No	Shared across target relations	FB + ClueWeb	MAP, MQ
CLUSTERPRA	Nodes	Vector	-	-	Random walk, cosine similarity	No	Yes	Yes	No	Separate for each	NELL, FB15k	MAP, MRR
CONV-DISTMULT	Vector	Bilinear map	Bilinear multiplication	Convolution over shared dependency edges	-	No	Yes	No	No	Shared parametrization via dep-edges	FB15k-237	MRR, HITS@10

Table 1: Summary