Incorporating Selectional Preferences in Multi-hop Relation Extraction

Rajarshi Das, Arvind Neelakantan, David Belanger and Andrew McCallum
College of Information and Computer Sciences
University of Massachusetts Amherst
{rajarshi, arvind, belanger, mccallum}@cs.umass.edu

Abstract

Relation extraction is one of the core challenges in automated knowledge base construction. One line of approach for relation extraction is to perform multi-hop reasoning on the paths connecting an entity pair to infer new relations. While these methods have been successfully applied for knowledge base completion, they do not utilize the entity or the entity type information to make predictions. In this work, we incorporate *selectional preferences*, i.e., relations enforce constraints on the allowed entity types for the candidate entities, to multi-hop relation extraction by including entity type information. We achieve a 17.67% improvement in MAP score in a relation extraction task when compared to a method that does not use entity type information.

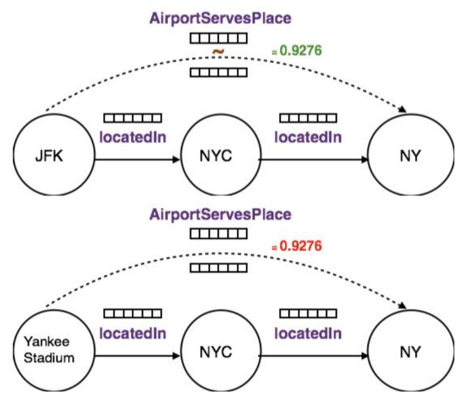

Figure 1: The two paths above consists of the same relations (`locatedIn` → `locatedIn`) and hence the model of Neelakantan (2015) will assign them the same score for the relation *AirportServesPlace* without considering the fact that *Yankee Stadium* is not an airport.

1 Introduction

Knowledge Bases (KB) sre structured knowledge sources widely used in applications like question answering (Kwiatkowski et al., 2013; Berant et al., 2013; Bordes et al., 2014) and search engines like *Google Search* and *Microsoft Bing*. This has led to the creation of large KBs like Freebase (Bollacker et al., 2008), YAGO (Suchanek et al., 2007) and NELL (Carlson et al., 2010). KBs contains millions of facts usually in the form of triples $(entity1, relation, entity2)$. However, KBs are woefully incomplete (Min et al., 2013) missing important facts, and hence limiting their usefulness in downstream tasks.

To overcome this difficulty, Knowledge Base Completion (KBC) methods aim to complete the KB using existing facts. For example, we can infer nationality of a person from their place of birth. A common approach in many KBC methods for relation extraction is reasoning on individual relations (single-hop reasoning) to predict new relations (Mintz et al., 2009; Bordes et al., 2013; Riedel et al., 2013; Socher et al., 2013). For example, predicting *Nationality(X, Y)* from *BornIn(X, Y)*. The performance of relation extraction methods have been greatly improved by incorporating *selectional preferences*, i.e., relations enforce constraints on the allowed entity types for the candidate entities, both in sentence level (Roth and Yih, 2007; Singh et al., 2013) and KB relation extraction (Chang et al., 2014).

Another line of work in relation extraction, performs reasoning on the paths (multi-hop reasoning on paths of length $\geq$ 1) connecting an entity pair (Lao et al., 2011; Lao et al., 2012; Gardner et al., 2013; Gardner et al., 2014; Neelakantan et al., 2015; Guu et al., 2015). For example, these models can infer the relation `PlaysInLeague(Tom Brady, NFL)` from the facts `PlaysForTeam(Tom Brady, New England Patriots)` and `PartOf(New England Patriots, NFL)`. All these methods utilize only the relations in the path and do not include any information about the entities.

In this work, we extend the method of Neelakantan (2015) by incorporating entity type information. Their method can generalize to paths unseen in training by composing embeddings of relations in the path non-linearly using a Recurrent Neural Network (RNN) (Werbos, 1990). While entity type information has been successfully incorporated into relation extraction methods that perform single hop reasoning, here, we include them for multi-hop relation extraction. For example, Figure 1 illustrates an example where reasoning without type information would score both the paths equally although the latter path should receive a lesser score since there is an entity type mismatch for the first entity. Our approach constructs vector representation of paths in the KB graph from representations of relations and entity types occurring in the path. We achieve a 17.67% improvement in Mean Average Precision (MAP) scores in a relation extraction task when compared to a method that does not use entity type information.

2 Model

This paper extends the Recurrent Neural Network model of Neelakantan (2015) by jointly reasoning over the relations and entity types occurring in the paths between an entity pair. Paths are represented as dense vectors formed by composing embeddings of relation and entity (or its types) occurring at each step. Figure 2 illustrates the encoder architecture for a path between an entity pair. The $[\cdot]$ in figure 2 denotes the concatenate operation. As will be described later, we try concatenating just the entity embeddings, its types or both.

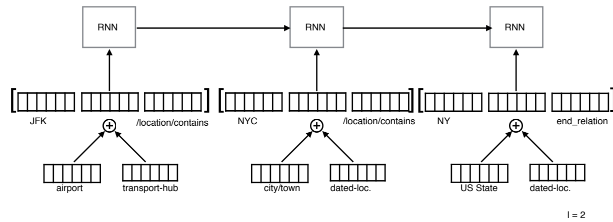

Figure 2: The encoder network for a path between an entity pair. The inputs to the network are embeddings of entities, entity types and relations. Also note we have a dummy relation token `end_relation` for the last entity of the path. In the network above, at each time step the entity embeddings are concatenated with the sum of its type embeddings, followed by the embeddings of the relation type and are fed as input to the recurrent network

The relation types considered in our work are either fixed symbolic types defined in the Freebase schema such as `/people/person/nationality` or a free text relation from Clueweb (Orr et al., 2013) such as `born in`. In Freebase, an entity is associated with several types. For example, the entity `Barack Obama` has types such as `President`, `Author` and `Award Winner`. We consider two different ways to represent the entity: a separate embedding for every entity and Let $v_e(e) \in \mathbb{R}^m$ denote the vector representation of an entity and taking the sum of embeddings of the top l (sorted in decreasing order of corpus frequency) types for an entity. l is a hyperparameter of the model.

Let $v_r(\delta) \in \mathbb{R}^d$ denote the vector representation of relation type δ. Let $v_e(e) \in \mathbb{R}^m$ denote the vector representation of an entity and $v_{et}(e) \in \mathbb{R}^n$ denote the vector representation of an entity obtained by taking the sum of its top l types. Let π be a path between the entity pair $(e1, e2)$ containing the relation types $\delta_1, \delta_2, \ldots, \delta_N$.

In the following section, we first briefly describe the model proposed by Neelakantan (2015) (RNN model henceforth) followed by our extensions to it.

2.1 RNN Model

The RNN model only considers the representations of relation type present in the path. More precisely, the vector representation $h_t \in \mathbb{R}^p$ of path

$\delta_1, \delta_2, \ldots, \delta_t$ $(1 \leq t \leq N)$ is computed recursively as

$$h_t = f\left(W_{hh} h_{t-1} + W_{rh} v_r\left(\delta_t\right)\right) \quad (1)$$

The vector representation of the entire path is h_N where N is the length of the path. Here $W_{h,h} \in \mathbb{R}^{p \times p}$ and $W_{rh} \in \mathbb{R}^{p \times d}$ are composition matrices between the previous step in the path and the relation vector at the current step respectively and f is a non-linear activation function.

Extension with entity (and types)

The previous model can be extended to incorporate the embeddings of entities along with relations occurring at each step in the path. We consider learning a separate representation for every entity and representing an entity using its entity types.

- RNN + Entity: In this model, we add the embedding of the entity.

$$h_t = f\left(W_{hh} h_{t-1} + W_{rh} v_r\left(\delta_t\right) + W_{eh} v_e\left(e_t\right)\right) \quad (2)$$

- RNN + Type: In this model, we add the embedding of the *entity* obtained from its types at each step.

$$h_t = f\left(W_{hh} h_{t-1} + W_{rh} v_r\left(\delta_t\right) + W_{th} v_{et}\left(e_t\right)\right) \quad (3)$$

- RNN + Entity + Type: In this model, we use both the representations of the entity.

$$\begin{aligned} h_t = f(&W_{hh} h_{t-1} + W_{rh} v_r\left(\delta_t\right) \\ &+ W_{eh} v_e\left(e_t\right) + W_{th} v_{et}\left(e_t\right)) \end{aligned} \quad (4)$$

Here e_t denotes the t^{th} entity occurring in the path between an entity pair and $W_{eh} \in \mathbb{R}^{p \times m}, W_{th} \in \mathbb{R}^{p \times n}$ are new composition matrices due to the entity and its types respectively. In all of our experiments f is the sigmoid activation function.

2.2 Model Training

We train a separate RNN model for each target relation. The parameters for each model are the embedding of the relations, entities and types, and the various composition matrices (as applicable) . They are trained to maximize the likelihood of the training data.The score of a path π w.r.t to the target relation δ is

$$score(\pi, \delta) = \sigma\left(v\left(\pi\right) \cdot v\left(\delta\right)\right) \quad (5)$$

We then choose the path which has the highest score similar to (Weston et al., 2013; Neelakantan et al., 2014). Selecting just one path (out of typically hundreds to thousands of paths) between entity pairs might lead to our model ignoring informative paths, especially during the initial stages of training. To alleviate this issue we also experiment by selecting the top k paths which have the highest score for a given entity pair and relation with the resultant score being the average of the top k scores.

3 Experiments & Results

In all of our experiments, we set the dimension of the relations, entity and their type embeddings to be 50. For a fair comparison with our model which has more number of parameters due to the entity and/or type embeddings, we experiment by varying the dimension of the relation embeddings between 50, 100 and 150 for the baseline model. We use Adam (Kingma and Ba, 2014) for optimization with the default hyperparameter settings. The models are trained for 15 epochs beyond which we observed overfitting on a held-out development set. We set $l = 7$ and $k = 5$ in our experiments. We experiment with 12 target relations.

3.1 Data

We run our experiments on the dataset released by Neelakantan el al. (2015) which is a subset of Freebase enriched with information from ClueWeb. The dataset comprises of entity pairs with a set of paths connecting them in the knowledge graph. However the paths had the entity information missing from them and only contained the relation types occurring in them. For example, consider the path $SatyaNadella \xrightarrow{ceoAt} Microsoft \xrightarrow{locatedIn} Seattle \xrightarrow{cityIn} Washington$. The original dataset had the entities in-between such as $Microsoft$ and $Seattle$ missing from it.

We augment the dataset with the entities present in them. To gather the entities we do a depth first traversal starting from the first entity of the entity pair and following the relation types until we reach the last entity of the pair. In cases of one-to-many

Stats	Full dataset	Current experiments
# test relations	46	12
# entity pairs	3.22M	839K
# entity pairs (train)	605K	161K
# entity pairs (test)	2M	533K
Avg. paths /relation	3.77M	3.43 M

Table 1: Statistics of the dataset

relations we choose the next entity to be traversed at random. Due to the combinatorial search space we limit the total number of edges traversed beyond which we ignore the path. Therefore the number of paths between an entity pair would be less than in the original dataset. However, we are continuously augmenting the dataset and we will make this dataset freely available to the research community[1]. Table 1 displays some statistics of the dataset gathered till now and also the subset that was used for running the current experiments.

3.2 Link Prediction

We compare our models with the baseline model on predicting whether an entity pair participates in a target relation. We rank the entity pairs in the test set based on their scores and calculate the Mean Average Precision (MAP) score for the ranking following previous work (Riedel et al., 2013; Neelakantan et al., 2015). Table 2 lists the MAP scores of both the models averaged over 12 freebase relation types.

Incorporating *selectional preference* by adding entity types gives a significant boost in scores (17.67 % over the baseline model.). However, we see a drop in performance on adding just entities. This is primarily because during test time we encounter a lot of previously unseen entities and hence we do not have learned embeddings for them. We overcome this problem by representing the entity using its observed types in Freebase. In future work, we would consider representing the entity additionally using context words (Yaghoobzadeh and Schütze, 2015).

Although considering top-k paths improves the performance of the baseline model, we observe that they provide almost similar scores with entity types. We run our experiments with $k = 5$ and we would report results after tuning k in the next version of the paper.

[1]On acceptance to the worksop, the url would be added in the camera-ready version.

Model		**MAP**
Max	RNN (50)	0.5991
	RNN (100)	0.6020
	RNN (150)	0.6272
	RNN + Entity	0.5593
	RNN + Entity + Type	0.5995
	RNN + Types	**0.7084**
Top-K	RNN (50)	0.6241
	RNN (100)	0.6184
	RNN (150)	0.6312
	RNN + Entity	0.5968
	RNN + Entity + Type	0.6322
	RNN + Types	**0.7014**

Table 2: Mean Average Precision scores averaged over 12 relations. The number in the parentheses denotes the dimension of the embedding of the relations type in the baseline model.

.

4 Conclusion

In this work, we incorporate *selectional preferences* to a multi-hop relation extraction method. We plan to release the dataset we collected for this project. We achieve a 17.67% improvement in MAP score in a relation extraction task when compared to a method that does not use entity type information.

References

Jonathan Berant, Vivek Srikumar, Pei-Chun Chen, Abby Vander Linden, Brittany Harding, Brad Huang, and Christopher D. Manning. 2013. Semantic parsing on freebase from question-answer pairs. In *Empirical Methods in Natural Language Processing.*

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 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08.

Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In Christopher J. C. Burges, Léon Bottou, Zoubin Ghahramani, and Kilian Q. Weinberger, editors, *NIPS*, pages 2787–2795.

Antoine Bordes, Sumit Chopra, and Jason Weston. 2014. Question answering with subgraph embeddings. In *Empirical Methods in Natural Language Processing.*

Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R. Hruschka, and Tom M. Mitchell. 2010. Toward an architecture for never-ending language learning. In *In AAAI*.

Kai-Wei Chang, Wen tau Yih, Bishan Yang, and Christopher Meek. 2014. Typed tensor decomposition of knowledge bases for relation extraction. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*.

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*.

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*.

K. Guu, J. Miller, and P. Liang. 2015. Traversing knowledge graphs in vector space. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Tom Kwiatkowski, Eunsol Choi, Yoav Artzi, and Luke. Zettlemoyer. 2013. Scaling semantic parsers with on-the-fly ontology matching. In *Empirical Methods in Natural Language Processing.*

Ni Lao, Tom Mitchell, and William W. Cohen. 2011. Random walk inference and learning in a large scale knowledge base. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 529–539, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ni Lao, Amarnag Subramanya, Fernando Pereira, and William W. Cohen. 2012. Reading the web with learned syntactic-semantic inference rules. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 1017–1026, Stroudsburg, PA, USA. Association for Computational Linguistics.

Bonan Min, Ralph Grishman, Li Wan, Chang Wang, and David Gondek. 2013. Distant supervision for relation extraction with an incomplete knowledge base. In *In Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL.*

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*.

Arvind Neelakantan, Jeevan Shankar, Alexandre Passos, and Andrew McCallum. 2014. Efficient non-parametric estimation of multiple embeddings per word in vector space. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1059–1069, Doha, Qatar, October. Association for Computational Linguistics.

Arvind Neelakantan, Benjamin Roth, and Andrew McCallum. 2015. Compositional vector space models for knowledge base completion. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 156–166, Beijing, China, July. Association for Computational Linguistics.

Dave Orr, Amar 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`.

Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M. Marlin. 2013. Relation extraction with matrix factorization and universal schemas. In *The North American Chapter of the Association for Computational Linguistics*.

Dan Roth and Wen-tau Yih. 2007. Global inference for entity and relation identification via a linear programming formulation. In *In Introduction to Statistical Relational Learning.*

Sameer Singh, Sebastian Riedel, Brian Martin, Jiaping Zheng, and Andrew McCallum. 2013. Joint inference of entities, relations, and coreference. In *CIKM Workshop on Automated Knowledge Base Construction (AKBC)*.

Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. 2013. Reasoning with neural tensor networks for knowledge base completion. In C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 926–934. Curran Associates, Inc.

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*, WWW '07.

P. Werbos. 1990. Backpropagation through time: what does it do and how to do it. In *Proceedings of IEEE*, volume 78.

Jason Weston, Ron Weiss, and Hector Yee. 2013. Nonlinear latent factorization by embedding multiple user interests. In *ACM International Conference on Recommender Systems (RecSys)*.

Yadollah Yaghoobzadeh and Hinrich Schütze. 2015. Corpus-level fine-grained entity typing using contextual information. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 715–725.