Improving Learning and Inference in a Large Knowledge-base using Latent Syntactic Cues

Matt Gardner, Partha Pratim Talukdar, Bryan Kisiel, and Tom Mitchell
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213, USA
{mg1,ppt,bkisiel,tom.mitchell}@cs.cmu.edu

Abstract

Automatically constructed Knowledge Bases (KBs) are often incomplete and there is a genuine need to improve their coverage. Path Ranking Algorithm (PRA) is a recently proposed method which aims to improve KB coverage by performing inference directly over the KB graph. For the first time, we demonstrate that addition of edges labeled with latent features mined from a large dependency parsed corpus of 500 million Web documents can significantly outperform previous PRA-based approaches on the KB inference task. We present extensive experimental results validating this finding. The resources presented in this paper are publicly available.

1 Introduction

Over the last few years, several large scale Knowledge Bases (KBs) such as Freebase (Bollacker et al., 2008), NELL (Carlson et al., 2010), and YAGO (Suchanek et al., 2007) have been developed. Each such KB consists of millions of facts (e.g., (*Tiger Woods*, *playsSport*, *Golf*)) spanning over multiple relations. Unfortunately, these KBs are often incomplete and there is a need to increase their coverage of facts to make them useful in practical applications.

A strategy to increase coverage might be to perform inference directly over the KB represented as a graph. For example, if the KB contained the following facts, (*Tiger Woods*, *participatesIn*, *PGA Tour*)) and (*Golf*, *sportOfTournament*, *PGA Tour*), then by putting these two facts together, we could potentially infer that (*Tiger Woods*, *playsSport*, *Golf*). The recently proposed Path Ranking Algorithm (PRA) (Lao and Cohen, 2010) performs such inference by automatically learning semantic inference rules over the KB (Lao et al., 2011). PRA uses features based off of sequences of edge types, e.g., ⟨*playsSport*, *sportOfTournament*⟩, to predict missing facts in the KB.

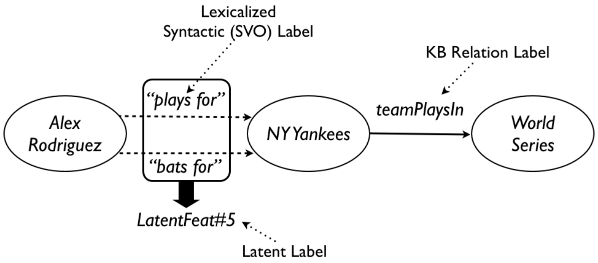

Figure 1: Example demonstrating how lexicalized syntactic edges can improve connectivity in the KB enabling PRA (Lao and Cohen, 2010) to discover relationships between *Alex Rodriguez* and *World Series*. Edges with latent labels can improve inference performance by reducing data sparsity. See Section 1.1 for details.

PRA was extended by (Lao et al., 2012) to perform inference over a KB augmented with dependency parsed sentences. While this opens up the possibility of learning syntactic-semantic inference rules, the set of syntactic edge labels used are just the *unlexicalized* dependency role labels (e.g., *nobj*, *dobj*, etc., without the corresponding words), thereby limiting overall expressitivity of the learned inference rules. To overcome this limitation, in this paper we augment the KB graph by adding edges with more expressive *lexicalized* syntactic labels (where the labels are words instead of dependencies). These additional edges, e.g., (*Alex Rodriguez*, *"plays for"*, *NY Yankees*), are mined by extracting 600 million Subject-Verb-Object (SVO) triples from a large corpus of 500m dependency parsed documents, which would have been prohibitively expensive to add directly as in (Lao et al., 2012). In order to overcome the explosion of path features and data sparsity, we derive edge labels by learning latent embeddings of the lexicalized edges. Through extensive experiments on real world datasets, we demonstrate effectiveness of the proposed approach.

1.1 Motivating Example

In Figure 1, the KB graph (only solid edges) is disconnected, thereby making it impossible for PRA to discover any relationship between *Alex Rodriguez* and *World Series*. However, addition of the two edges with SVO-based lexicalized syntactic edges (e.g., (*Alex Rodriguez*, *plays for*, *NY Yankees*)) restores this inference possibility. For example, PRA might use the edge sequence ⟨*"plays for"*, *teamPlaysIn*⟩ as evidence for predicting the relation instance (*Alex Rodriguez*, *athleteWonChampionship*, *World Series*). Unfortunately, such naïve addition of lexicalized edges may result in significant data sparsity, which can be overcome by mapping lexicalized edge labels to some latent embedding (e.g., (*Alex Rodriguez*, *LatentFeat#5*, *NY Yankees*) and running PRA over this augmented graph. Using latent embeddings, PRA could then use the following edge sequence as a feature in its prediction models: ⟨*LatentFeat#5*, *teamPlaysIn*⟩. We find this strategy to be very effective as described in Section 4.

2 Related Work

There is a long history of methods using suface-level lexical patterns for extracting relational facts from text corpora (Hearst, 1992; Brin, 1999; Agichtein and Gravano, 2000; Ravichandran and Hovy, 2002; Etzioni et al., 2004). Syntactic information in the form of dependency paths have been explored in (Snow et al., 2006; Suchanek et al., 2006). A method of latent embedding of relation instances for sentence-level relation extraction was shown in (Wang et al., 2011). However, none of this prior work makes explicit use of the background KBs as we explore in this paper.

Path Ranking Algorithm (PRA) (Lao and Cohen, 2010) has been used previously to perform inference over graph-structured KBs (Lao et al., 2011), and to learn formation of online communities (Settles and Dow, 2013). In (Lao et al., 2012), PRA is extended to perform inference over a KB using syntactic information from parsed text. In contrast to these previous PRA-based approaches where all edge labels are either KB labels or at surface-level, in this paper we explore using latent edge labels in addition to surface-level labels in the graph over which PRA is applied. In particular, we focus on the problem of performing inference over a large KB and learn latent edge labels by mining dependency syntax statistics from a large text corpus.

Though we use Principal Components Analysis (PCA) for dimensionality reduction for the experiments in this paper, this is by no means the only choice. Various other dimensionality reduction techniques, and in particular, other verb clustering techniques (Korhonen et al., 2003), may also be used.

OpenIE systems such as Reverb (Etzioni et al., 2011) also extract verb-anchored dependency triples from large text corpus. In contrast to such approaches, we focus on how latent embedding of verbs in such triples can be combined with explicit background knowledge to improve coverage of existing KBs. This has the added capability of inferring facts which are not explicitly mentioned in text.

The recently proposed Universal Schema (Riedel et al., 2013) also demonstrates the benefit of using latent features for increasing coverage of KBs. Key differences between that approach and ours include our use of syntactic information as opposed to surface-level patterns in theirs, and also the ability of the proposed PRA-based method to generate useful inference rules which is beyond the capability of the matrix factorization approach in (Riedel et al., 2013).

3 Method

3.1 Path Ranking Algorithm (PRA)

In this section, we present a brief overview of the Path Ranking Algorithm (PRA) (Lao and Cohen, 2010), building on the notations in (Lao et al., 2012). Let $G = (V, E, T)$ be the graph, where V is the set of vertices, E is the set of edges, and T is the set of edge types. For each edge $(v_1, t, v_2) \in E$, we have

$v_1, v_2 \in V$ and $t \in T$. Let $R \subset T$ be the set of types predicted by PRA. R could in principal equal T, but in this paper we restrict prediction to KB relations, while T also includes types derived from surface text and latent embeddings. Let $\pi = \langle t_1, t_2, \ldots, t_w \rangle$ be a *path type* of length w over graph G, where $t_i \in T$ is the type of the i^{th} edge in the path. Each such path type is also a *feature* in the PRA model. For a given source and target node pair $s, t \in V$, let $P(s \to t; \pi)$ be the value of the feature π specifying the probability of reaching node t starting from node s and following a path constrained by path type π. We approximate these probabilities using random walks. A value of 0 indicates unreachability from s to t using path type π.

Let $B = \{\pi_1, \ldots, \pi_m\}$ be the set of all features (path types). The score that relation r holds between node s and node t is given by the following function:

$$\text{Score}_{\text{PRA}}(s, t, r) = \sum_{\pi \in B} P(s \to t; \pi)\, \theta^r_\pi$$

where θ^r_π is the weight of feature π in class $r \in R$.

Feature Selection: The set B of possible path types grows exponentially in the length of the paths that are considered. In order to have a manageable set of features to compute, we first perform a feature selection step. The goal of this step is to select for computation only those path types that commonly connect sources and targets of relation r. We perform this feature selection by doing length-bounded random walks from a given list of source and target nodes, keeping track of how frequently each path type leads from a source node to a target node. The most common m path types are selected for the set B.

Training: We perform standard logistic regression with L2 regularization to learn the weights θ^r_π. We follow the strategy in (Lao and Cohen, 2010) to generate positive and negative training instances.

3.2 $\text{PRA}_{\text{syntactic}}$

In this section, we shall extend the knowledge graph $G = (V, E, T)$ from the previous section with an augmented graph $G^{'} = (V, E^{'}, T^{'})$, where $E \subset E^{'}$ and $T \subset T^{'}$, with the set of vertices unchanged.

In order to get the edges in $E^{'} - E$, we first collect a set of Subject-Verb-Object (SVO) triples $D = \{(s, v, o, c)\}$ from a large dependency parsed text corpus, with $c \in \mathbb{R}_+$ denoting the frequency of this triple in the corpus. The additional edge set is then defined as $E_{\text{syntactic}} = E^{'} - E = \{(s, v, o) \mid \exists (s, v, o, c) \in D,\ s, o \in V\}$. We define $S = \{v \mid \exists (s, v, o) \in E_{\text{syntactic}}\}$ and set $T^{'} = T \cup S$. In other words, for each pair of directly connected nodes in the KB graph G, we add an additional edge between those two nodes for each verb which takes the NPs represented by two nodes as subjects and objects (or vice versa) as observed in a text corpus. In Figure 1, (*Alex Rodriguez*, *"plays for"*, *NY Yankees*) is an example of such an edge.

PRA is then applied over this augmented graph $G^{'}$, over the same set of prediction types R as before. We shall refer to this version of PRA as $\text{PRA}_{\text{syntactic}}$. For the experiments in this paper, we collected $|D| = 600$ million SVO triples[1] from the entire ClueWeb corpus (Callan et al., 2009), parsed using the Malt parser (Nivre et al., 2007) by the Hazy project (Kumar et al., 2013).

3.3 $\text{PRA}_{\text{latent}}$

In this section we construct $G^{''} = (V, E^{''}, T^{''})$, another syntactic-information-induced extension of the knowledge graph G, but instead of using the surface forms of verbs in S (see previous section) as edge types, we derive those edges types $T^{''}$ based on latent embeddings of those verbs. We note that $E \subset E^{''}$, and $T \subset T^{''}$.

In order to learn the latent or low dimensional embeddings of the verbs in S, we first define $Q_S = \{(s, o) \mid \exists (s, v, o, c) \in D,\ v \in S\}$, the set of subject-object tuples in D which are connected by at least one verb in S. We now construct a matrix $X_{|S| \times |Q_S|}$ whose entry $X_{v,q} = c$, where $v \in S, q = (s, o) \in Q_S$, and $(s, v, o, c) \in D$. After row normalizing and centering matrix X, we apply PCA on this matrix. Let $A_{|S| \times d}$ with $d << |Q_S|$ be the low dimensional embeddings of the verbs in S as induced by PCA. We use two strategies to derive mappings for verbs from matrix A.

- $\text{PRA}_{\text{latent}_c}$: The verb is mapped to concatenation of the $\frac{k}{2}$ most positive columns in the row in A that corresponds to the verb. Similarly, for the most negative $\frac{k}{2}$ columns.

[1] This data and other resources from the paper are publicly available at http://rtw.ml.cmu.edu/emnlp2013_pra/.

	Precision	Recall	F1
PRA	0.800	0.331	0.468
$\mathrm{PRA}_{\mathrm{syntactic}}$	0.804	0.271	0.405
$\mathrm{PRA}_{\mathrm{latent}_c}$	**0.885**	0.334	0.485
$\mathrm{PRA}_{\mathrm{latent}_d}$	0.868	**0.424**	**0.570**

Table 1: Comparison of performance of different variants of PRA micro averaged across 15 NELL relations. We find that use of latent edge labels, in particular the proposed approach $\mathrm{PRA}_{\mathrm{latent}_d}$, significantly outperforms other approaches. This is our main result. (See Section 4)

- $\mathrm{PRA}_{\mathrm{latent}_d}$: The verb is mapped to disjunction of top-k most positive and negative columns in the row in A that corresponds to the verb.

4 Experiments

We compared the various methods using 15 NELL relations. For each relation, we split NELL's known relation instances into 90% training and 10% testing. For each method, we then selected 750 path features and trained the model, as described in Section 3, using GraphChi (Kyrola et al., 2012) to perform the random walk graph computations. To evaluate the model, we took all source nodes in the testing data and used the model to predict target nodes. We report the precision and recall (on the set of known target nodes) of the set of predictions for each model that are above a certain confidence threshold. Because we used strong regularization, we picked for our threshold a model score of 0.405, corresponding to 60% probability of the relation instance being true; values higher than this left many relations without any predictions. Table 1 contains the results.

As can be seen in the table, $\mathrm{PRA}_{\mathrm{syntactic}}$ on average performs slightly worse than PRA. While the extra syntactic features are very informative for some relations, they also introduce a lot of sparsity, which makes the model perform worse on other relations. When using latent factorization methods to reduce the sparsity of the syntactic features, we see a significant improvement in performance. $\mathrm{PRA}_{\mathrm{latent}_c}$ has a 45% reduction in precision errors vs. PRA while maintaining the same recall, and $\mathrm{PRA}_{\mathrm{latent}_d}$ reduces precision errors by 35% while improving recall by 27%. Section 4.1 contains some qualitative analysis of how sparsity is reduced with the latent methods. As a piece quantitative analysis, there were 908 possible path types found in the feature selection step with PRA on the relation *cityLiesOnRiver* (of which we then selected 750). For $\mathrm{PRA}_{\mathrm{syntactic}}$, there were 73,820, while $\mathrm{PRA}_{\mathrm{latent}_c}$ had 47,554 and $\mathrm{PRA}_{\mathrm{latent}_d}$ had 58,414.

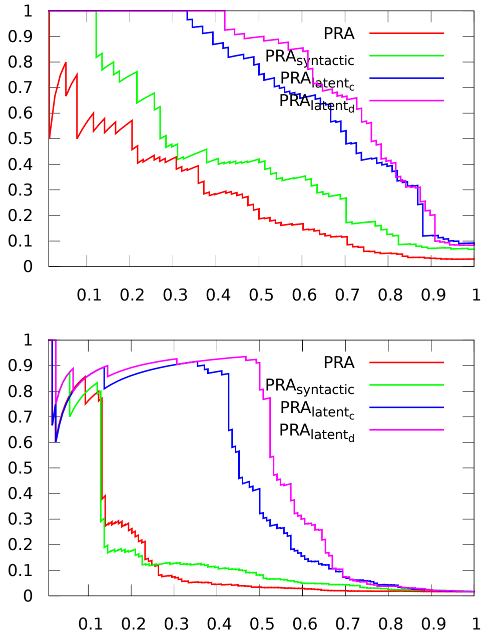

Figure 2: Precision (y axis) - Recall (x axis) plots for the relations *cityLiesOnRiver* (top) and *athletePlaysForTeam* (bottom). $\mathrm{PRA}_{\mathrm{latent}_d}$ (rightmost plot), the proposed approach which exploits latent edge labels, outperforms other alternatives.

Table 2 shows F1 scores for each model on each relation, and Figure 2 shows representative Precision-Recall plots for two NELL relations. In both cases, we find that $\mathrm{PRA}_{\mathrm{latent}_d}$ significantly outperforms other baselines.

4.1 Discussion

While examining the model weights for each of the methods, we saw a few occasions where surface relations and NELL relations combined to form interpretable path types. For example, in *athletePlaysForTeam*, some highly weighted features took the form of ⟨*athletePlaysSport*, *"(sport) played by (team)"*⟩. A high weight on this feature would bias the prediction towards teams that are known to play the same sport as the athlete.

For PRA, the top features for the best performing relations are path types that contain a single edge

	PRA	$\mathrm{PRA}_{\mathrm{syntactic}}$	$\mathrm{PRA}_{\mathrm{latent}_c}$	$\mathrm{PRA}_{\mathrm{latent}_d}$
animalIsTypeOfAnimal	0.52	0.50	0.47	**0.53**
athletePlaysForTeam	0.22	0.21	0.56	**0.64**
athletePlaysInLeague	**0.81**	0.75	0.73	0.74
cityLiesOnRiver	0.05	0	0.07	**0.31**
cityLocatedInCountry	0.15	0.20	0.45	**0.55**
companyCeo	0.29	0.18	0.25	**0.35**
countryHasCompanyOffice	0	0	0	0
drugHasSideEffect	**0.96**	0.95	0.94	0.94
headquarteredIn	0.31	0.11	0.41	**0.64**
locationLocatedWithinLocation	0.40	0.38	0.38	**0.41**
publicationJournalist	0.10	0.06	0.10	**0.16**
roomCanContainFurniture	0.72	0.70	0.71	**0.73**
stadiumLocatedInCity	0.53	0	0.13	**0.67**
teamPlaysAgainstTeam	**0.47**	0.24	0.26	0.21
writerWroteBook	0.59	0.62	0.73	**0.80**

Table 2: F1 performance of different variants of PRA for all 15 relations tested.

which is a supertype or subtype of the relation being predicted. For instance, for the relation *athletePlaysForTeam* (shown in Figure 2), the highest-weighted features in PRA are *athleteLedSportsTeam* (more specific than *athletePlaysForTeam*) and *personBelongsToOrganization* (more general than *athletePlaysForTeam*). For the same relation, $\mathrm{PRA}_{\mathrm{syntactic}}$ has features like *"scored for"*, *"signed"*, *"have"*, and *"led"*. When using a latent embedding of these verb phrases, *"signed"*, *"have"*, and *"led"* all have the same representation in the latent space, and so it seems clear that $\mathrm{PRA}_{\mathrm{latent}}$ gains a lot by reducing the sparsity inherent in using surface verb forms.

For *cityLiesOnRiver*, where PRA does not perform as well, there is no NELL relation that is an immediate supertype or subtype, and so PRA does not have as much evidence to use. It finds features that, e.g., are analogous to the statement "cities in the same state probably lie on the same river". Adding lexical labels gives the model edges to use like *"lies on"*, *"runs through"*, *"flows through"*, *"starts in"* and *"reaches"*, and these features give a significant boost in performance to $\mathrm{PRA}_{\mathrm{syntactic}}$. Once again, almost all of those verb phrases share the same latent embedding, and so $\mathrm{PRA}_{\mathrm{latent}}$ gains another significant boost in performance by combining them into a single feature.

5 Conclusion

In this paper, we introduced the use of latent lexical edge labels for PRA-based inference over knowledge bases. We obtained such latent edge labels by mining a large dependency parsed corpus of 500 million web documents and performing PCA on the result. Through extensive experiments on real datasets, we demonstrated that the proposed approach significantly outperforms previous state-of-the-art baselines.

Acknowledgments

We thank William Cohen (CMU) for enlightening conversations on topics discussed in this paper. We thank the ClueWeb project (CMU) and the Hazy Research Group (http://hazy.cs.wisc.edu/hazy/) for their generous help with data sets; and to the anonymous reviewers for their constructive comments. This research has been supported in part by DARPA (under contract number FA8750-13-2-0005), and Google. Any opinions, findings, conclusions and recommendations expressed in this paper are the authors' and do not necessarily reflect those of the sponsors.

References

Eugene Agichtein and Luis Gravano. 2000. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the Fifth ACM conference on Digital libraries*.

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 SIGMOD*.

Sergey Brin. 1999. Extracting patterns and relations from the world wide web.

J. Callan, M. Hoy, C. Yoo, and L. Zhao. 2009. Clueweb09 data set. *boston.lti.cs.cmu.edu*.

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

Oren Etzioni, Michael Cafarella, Doug Downey, Stanley Kok, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S Weld, and Alexander Yates. 2004. Web-scale information extraction in knowitall:(preliminary results). In *Proceedings of WWW*.

Oren Etzioni, Anthony Fader, Janara Christensen, Stephen Soderland, and Mausam Mausam. 2011. Open information extraction: The second generation. In *Proceedings of IJCAI*.

Marti A Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational Linguistics*.

Anna Korhonen, Yuval Krymolowski, and Zvika Marx. 2003. Clustering polysemic subcategorization frame distributions semantically. In *Proceedings of ACL*.

Arun Kumar, Feng Niu, and Christopher Ré. 2013. Hazy: making it easier to build and maintain big-data analytics. *Communications of the ACM*, 56(3):40–49.

Aapo Kyrola, Guy Blelloch, and Carlos Guestrin. 2012. Graphchi: Large-scale graph computation on just a pc. In *Proceedings of the 10th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, pages 31–46.

Ni Lao and William W Cohen. 2010. Relational retrieval using a combination of path-constrained random walks. *Machine learning*, 81(1):53–67.

Ni Lao, Tom Mitchell, and William W Cohen. 2011. Random walk inference and learning in a large scale knowledge base. In *Proceedings of EMNLP*. 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 EMNLP-CoNLL*.

J. Nivre, J. Hall, J. Nilsson, A. Chanev, G. Eryigit, S. Kübler, S. Marinov, and E. Marsi. 2007. Maltparser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13(02).

Deepak Ravichandran and Eduard Hovy. 2002. Learning surface text patterns for a question answering system. In *Proceedings of ACL*.

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

Burr Settles and Steven Dow. 2013. Let's get together: the formation and success of online creative collaborations. In *Proceedings of CHI*.

Rion Snow, Daniel Jurafsky, and Andrew Y Ng. 2006. Semantic taxonomy induction from heterogenous evidence. In *Proceedings of ACL*.

Fabian M Suchanek, Georgiana Ifrim, and Gerhard Weikum. 2006. Combining linguistic and statistical analysis to extract relations from web documents. In *Proceedings of KDD*.

Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *Proceedings of WWW*.

Chang Wang, James Fan, Aditya Kalyanpur, and David Gondek. 2011. Relation extraction with relation topics. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1426–1436. Association for Computational Linguistics.