Combining Vector Space Embeddings with Symbolic Logical Inference over Open-Domain Text

Matt Gardner*
Carnegie Mellon University
mg1@cs.cmu.edu

Partha Talukdar
Indian Institute of Science
ppt@serc.iisc.in

Tom Mitchell
Carnegie Mellon University
tom@cs.cmu.edu

Abstract

We have recently shown how to combine random walk inference over knowledge bases with vector space representations of surface forms, improving performance on knowledge base inference. In this paper, we formalize the connection of our prior work to logical inference rules, giving some general observations about methods for incorporating vector space representations into symbolic logic systems. Additionally, we present some promising preliminary work that extends these techniques to learning open-domain relations for the purpose of answering multiple choice questions, achieving 67% accuracy on a small test set.

Introduction

Recent advances in constructing large knowledge bases (KBs) have spurred the development of scalable tools for performing symbolic logical inference. One such tool, the Path Ranking Algorithm (PRA), uses random walks to find horn clause rules that are potentially useful for predicting new instances of a KB relation, then combines them with logistic regression to create a discriminatively trained logical inference model (Lao and Cohen, 2010; Lao, Mitchell, and Cohen, 2011). When used with open-domain relations derived from surface text, however, this technique suffers from the sparsity of textual representations—there are a lot of ways to express the same relationship in text, and they are each treated as separate symbols by PRA. We have recently introduced techniques to overcome this sparsity, first by using a clustering algorithm over a vector space representation of the surface forms (Gardner et al., 2013), and second by using vector space representations directly in the random walks performed by PRA (Gardner et al., 2014). We showed that these are effective means of combining scalable, symbolic logical inference with distributed representations of surface forms.

In this paper we present two main contributions. First, we formalize the connections between our recent work and logical inference, giving insight into how future work might better make use of distributed or neural representations of surface text in symbolic logic. Second, we present preliminary work on applying this kind of logical inference to a new problem. Previously, PRA has only been used to infer new instances of knowledge base relations. We show that these techniques are also useful in an open-domain multiple choice question answering task, where the goal of the system is to rank a set of open-domain relation triples for each possible answer. On a small set of 24 questions, we achieve an accuracy of 67%, on par with the state-of-the-art Aristo 1.0 system (Clark, Harrison, and Balasubramanian, 2013).

PRA as logical inference

Many systems exist for finding weighted horn clause rules from a set of relations (Schoenmackers et al., 2010; Quinlan, 1990). Given a set of inference rules, a set of relations, and a query, a common technique for performing logical inference is to construct a Markov logic network and use belief propagation or Monte Carlo techniques to answer the query (e.g., the Holmes system of Schoenmackers, Etzioni, and Weld (2008)). As the number of inference rules grows, however, performing inference in this manner can be quite expensive.

PRA, by contrast, both finds potential horn clause rules and performs inference over them with simple random walks over a graph representation of the knowledge base. PRA leverages the fact that known instances of a particular relation can be treated as training examples for learning a discriminatively trained model. The inference is impoverished in two ways, however: (1) the random walks can only find horn clause rules whose antecedents correspond to chains of edges in the graph, so the set of possible rules is restricted; and (2) the logistic regression model is not recursive; that is, all intermediate facts necessary for a proof must already be present in the knowledge base for PRA to make a correct inference.[1]

We can formalize the connection between logical inference and PRA as follows. For each relation R in the knowledge base, PRA finds a set of N horn clause rules, each of

* Research carried out in part while at the Allen Institute for Artificial Intelligence, Seattle, WA.

[1] This second issue can be mitigated by running PRA repeatedly, adding high-confidence predictions to the knowledge base after each run, as is currently done in the NELL system (Carlson et al., 2010). Additionally, ProPPR (Wang, Mazaitis, and Cohen, 2013) uses similar techniques to PRA that are also recursive.

which can be written as:

$$R(X,Y) \impliedby \bigwedge_{i=0}^{k} P_i(Z_i, Z_{i+1})$$

where k is the number of literals in the antecedent of the horn clause rule, $Z_0 = X$, $Z_{k+1} = Y$, and all other Z_i are free variables in the horn clause.

For a particular query $R(X,Y)$, we denote the value of each inference rule n with $\pi_n(X,Y)$, where $\pi_n(X,Y)$ is 1 if all of the $P_i(Z_i, Z_{i+1})$ facts in the rule are present in the knowledge base, and 0 otherwise.[2] Instead of creating a joint (generative) model over all facts in the knowledge base and assigning a probability to each horn clause rule, PRA creates a discriminative model for each relation and assigns a weight to each horn clause rule using logistic regression. Thus the potentially intractable problem of combining probabilities from independently created horn clause rules is avoided, and PRA assigns a probability to $R(X,Y)$ as follows:

$$\Pr(R(X,Y)) = \sigma\left(\sum_n w_n \pi_n(X,Y)\right)$$

where σ is the sigmoid function and w_n is the logistic regression weight for horn clause rule n.

Logical inference with vector space semantics

While the connection between the original PRA algorithm and logical inference is relatively straightforward and already known, our extensions of PRA to incorporate vector space semantics make the connection less obvious. Here we make that connection explicit, showing how our changes to the algorithm in our recent work map to changes in the formulas for logical inference above. We believe this mapping is potentially instructive, showing how other logical inference methods, such as Markov logic networks, could incorporate similar modifications to make use of vector space semantics.

First we take the simpler case of clustered surface forms (Gardner et al., 2013). A clustering defines a function, $C(P_i, P_j)$, which returns true if P_i and P_j share the same cluster. With this function, we can rewrite the formula for each horn clause inference rule as follows:

$$R(X,Y) \impliedby \bigwedge_{i=0}^{k} \left(\bigvee_{S_i \ s.t. \ C(P_i,S_i)} S_i(Z_i, Z_{i+1}) \right)$$

The rest of the PRA algorithm remains the same. Thus this clustering technique simply adds disjunctions to every horn clause rule containing a predicate in a cluster of size larger than 1, where the clusters are constructed using some distributed representation of the predicates. This will help in instances when sparsity of training data leads to some predicates in a horn clause rule only being seen at test time, and it will hurt when the addition of a disjunction clause to the rule decreases the rule's specificity (and thus its inferential utility). In our experiments, we found this technique to improve performance when surface forms were clustered, but not when knowledge base relations were clustered. This makes intuitive sense, as knowledge base relations already have a well-defined semantics in the context of the KB (and so adding a disjunction will only dilute the rule), while textual relations do not. It is possible that in a graph that combines multiple KBs (e.g., NELL, Freebase, YAGO, ConceptNet, etc.), using vector space representations for KB relations (and disjunctions generated from them) could improve performance.

The vector space random walk technique we recently introduced (Gardner et al., 2014) is slightly more complicated to formulate in logical notation, but still reduces essentially to the introduction of disjunctions in the horn clause rules. This technique modifies the random walks in PRA to follow an edge in the graph at node Z_i with probability proportional to the vector space similarity between the edge and the horn clause predicate P_i. Thus instead of a clustering function $C(P_i, P_j)$, we define a similarity function $\mathrm{Sim}(P_i, P_j)$, which assigns a score (typically) between -1 and 1 to any relation pair in the knowledge base.[3] With this function, our disjunction now includes *all* relations in the knowledge base for every predicate, but the probability of sampling a 1 for $\pi_n(X,Y)$ depends on the similarity function. Thus

$$R(X,Y) \impliedby \bigwedge_{i=0}^{k} \left(\bigvee_{S \in \mathrm{KB}} S(Z_i, Z_{i+1}) \right)$$

and

$$\Pr(\pi_n(X,Y)) \propto \prod_{i=0}^{k} \left(\sum_{S \ s.t. \ S(Z_i,Z_{i+1})} \exp\left(\mathrm{Sim}(P_i, S)\right) \right)$$

In conclusion, our use of vector space semantics to augment logical inference translates essentially to the introduction of disjunctions that are dependent in some way on similarity in the vector space (either through an explicit clustering, or through a modification to the sampling probability in PRA). Despite their simplicity, we showed these techniques to give significant improvements in performance when reasoning with surface forms (Gardner et al., 2013, 2014). This suggests that other methods of performing logical inference could also benefit from the targeted addition of disjunctions in inference rules, where these disjunctions are determined by some function over the vector space representations of relations.

Open-domain question answering with PRA

PRA has so far only been used to learn models of knowledge base predicates, and whether these techniques still work

[2] In practice, PRA only samples from these chains in the graph, so $\pi_n(X,Y)$ may be 0 when the facts are actually present. Additionally, previous work in PRA has used random walk probabilities for $\pi_n(X,Y)$; our experimentation has shown that binarizing these values makes little difference and may actually give better performance, so we use binary values here to simplify the discussion.

[3] If either P_i or P_j does not have a vector space representation, the function returns $-\infty$.

when applied to less semantically crisp open-domain relations has been an open question—the random walk technique makes heavy use of the domain and range constraints available for knowledge base predicates to filter out impossible predictions. Here we present some preliminary work on learning PRA models for open-domain relations in the context of a multiple choice question answering system. We show that a naive approach to formulating question answering into a PRA model achieves 67% accuracy on a small test set, on par with the state-of-the-art Aristo 1.0 system (Clark, Harrison, and Balasubramanian, 2013). This result gives us evidence that the discriminative logical inference described above is applicable to more than just inferring new edges in a symbolic knowledge base; we can also successfully reason over surface forms found in text.

Multiple choice question answering

Knowledge base inference with PRA relies on the fact that KB relations have domains and ranges to restrict the predictions made by the algorithm. Open-domain relations found in text have no such restrictions, and in our experimentation we have found that PRA performance suffers because of this when trying to learn models of these relations. Multiple choice question answering provides a nice stepping stone to move from KB relations to open-domain relations, because the task is simply to *rank* a small set of relation triples obtained from the multiple answers to the question, instead of producing a probability for any possible relation triple found in open-domain text. Successfully solving this task with PRA should provide insight into the modifications necessary to learn more general models of open-domain relations.

The questions we experimented with come from the New York Regents Exam (Clark, Harrison, and Balasubramanian, 2013). A couple of example questions follow:

> The functions of a plant's roots are to support the plant and
>
> (A) make food
>
> (B) produce fruit
>
> (C) take in water and nutrients
>
> (D) aid in germination
>
> Which tool should a student use to compare the masses of two small rocks?
>
> (A) balance
>
> (B) hand lens
>
> (C) ruler
>
> (D) measuring cup

We hand selected 24 of these questions that we believed could be answered by ranking relation triples, then manually constructed a set of triples for each answer. The triples corresponding to the first question above are as follows:

(A) (roots, make, food)

(B) (roots, produce, fruit)

(C) (roots, take in, water), (roots, take in, nutrients)

(D) (roots, aid in, germination)

Given this set of (subject, verb phrase, object) triples from the question set, we learn PRA models for each verb phrase. Given the model, we score each triple, then pick the answer with the highest combined score.

Experimental setup

To construct a graph for use with PRA, we used a set of automatically extracted SVO triples from several corpora: the texts used by the Aristo system (including the Barrons 4th Grade Science Study Guide, the CK12 Biology Textbook, and the science section of Simple English Wikipedia; below referred to as the "Aristo" extractions), and a subset of the SVO triples found in the ClueWeb corpus Talukdar, Wijaya, and Mitchell (2012) that were deemed relevant to the task (by sharing at least one of the three arguments in the relation triple; referred to below as the "ClueWeb" extractions). To find training data, we simply took all examples in the extractions that had the same relation; i.e., the training data for the PRA model for "make" (used to score the triple (roots, make, food)) consisted of all (subject, verb phrase, object) triples where the verb phrase was "make". We subsampled these triples until we had at most 3,500 examples per relation.

Our initial attempt used the triples in the questions and in the extractions exactly as found. However, as each noun phrase is represented as a node in the graph, long noun phrases in the triples made for a very disconnected graph, and PRA was not able to successfully score very many of the triples in the questions (e.g., "the masses of two small rocks", in the second example question above, had no connecting edges in the graph, and thus PRA could not give a score other than 0 for queries involving it). We tried to fix this issue by including edges between noun phrases: an edge from each noun phrase node to the node representing its head, edges from single-word noun phrases to their lemmas, and other similar edges. However, this made the graph so densely connected that random walk inference was intractable—the computation required to find connections by random walks is $O(\text{degree}^{\text{pathlength}})$, and adding edges in this way both dramatically increased the average degree of each node in the graph and significantly increased the path length between noun phrases in our training and test sets.

We ended up processing all of the triples (both from the questions and from the extractions) to only keep the lemmatized head of each noun phrase, and the lemmatized verb phrase (dropping any auxiliary or modal verbs, but keeping prepositions). For example, the triple (an offspring, can inherit, blue eyes) became (offspring, inherit, eye). This naively throws away a lot of information, some of which may be very important to answering the question (e.g., "scratched eyes" and "blue eyes" have the same simplification, though only one of them can be inherited). However, some method for making random walk inference both possible and tractable was necessary, and this naive approach gave us a starting place that actually had decent performance.

The experiments we present involved using PRA with a

Method	Correct	Precision	Recall
Best PRA result	16	0.70	0.67
Aristo	16	0.70	0.67
Baseline	5	0.83	0.21

Table 1: Number of correct answers, precision, and recall of the systems we experimented with, on our small test set of 24 questions.

graph constructed from the Aristo and ClueWeb extractions. For comparison, we include the Aristo system Clark, Harrison, and Balasubramanian (2013) and a naive baseline that scored each question triple by returning the number of its occurrences in the set of extractions.

Results and discussion

In Table 1 we show the results of our experiments. As can be seen in the table, our best PRA model had performance that was equivalent to that obtained by Aristo. We want to emphasize here that there was a lot of variance in our experiments—some runs of the system got as low as 11 correct, and attempts to use vector space random walks or additional edges from curated knowledge bases like NELL, Freebase or ConceptNet proved inconclusive. We need a larger test set to be confident in making a fair comparison between methods on this task. Our intent in presenting these results is simply to show that even a very naive approach to using PRA for question answering has the potential to match state-of-the-art performance. We believe that smarter approaches to training data selection and graph construction could give performance better than the current state-of-the-art, and we are actively pursuing research in this direction. In the constrained environment of multiple choice question answering, PRA can indeed reason successfully over open-domain relations.

In the remainder of this section we give some analysis of the successes and failures of PRA on this task. First, we show some interesting horn clauses that were assigned high weight by PRA; the algorithm is frequently able to discover important aspects of the semantics of textual relations. Because of space considerations, we only show examples for the relation "depend on":

- <"depend on", "depend on"> ("depend on" is transitive)
- <"use", "depend on">, and <"use", "give"$^{-1}$> (if I use something, I depend on its dependencies, or something that gives it)
- <"depend on", "make"> (if I depend on something, it might be because I depend on something it makes)

Next, we discuss some specific examples from the test set. First, note that the baseline (which simply looked up triples in the KB), only gave an answer for 6 of the questions. On 5 of them it was right, so the baseline has a high precision, but PRA is able to correctly answer many more questions because it can infer facts that are not present in the KB, or facts that use similar but slightly different verb phrases (e.g., (rain, is, precipitation) never occurred in the data, but (precipitation, consists of, rain) did, and PRA used that fact to correctly answer one of the questions). Interestingly, PRA correctly answered the question that the baseline gave a wrong answer for (the first example question given above), because even though the triples (root, make, food), and (root, produce, fruit) occurred several times in the extracted data, the learned PRA models assigned them low probability.

One weakness in our approach can be seen in the following question: A decomposer is an organism that (A) hunts and eats animals (B) migrates for the winter (C) breaks down dead plants and animals (D) uses water and sunlight to make food. The correct answer is C, but PRA gives B for the answer because (?, migrates for, winter) has a high prior probability in the model for "migrates for" (if not for that high prior, PRA would have gotten the question correct). A significant portion of the precision errors made by PRA would be solved with a better method for comparing scores across relations, especially where one of the arguments to the relation has an exceptionally high prior.

Finally, a very significant weakness with the approach we took has already been mentioned: only keeping the head of each noun phrase can throw away information that is critical to answering the question. Between "metal fork" and "plastic spoon", it is clear which is the better conductor of electricity, for example, but given only the heads "fork" and "spoon", there is no way to reliably obtain the correct answer. We did this stemming to decrease the computational requirements of PRA; our approach is essentially equivalent to clustering the nodes in the KB graph, shortening the connecting paths between noun phrases and decreasing the average degree of the graph. Another approach that does not throw away potentially critical information would be to prune the graph, keeping only edges that are deemed relevant to solving a particular query. This kind of "micro-KB" construction would also decrease the average degree of the graph and allow for deeper inferences while keeping the computational requirements of PRA to a manageable level.

Conclusion

Our recent work has shown how to combine vector space semantics with symbolic logical inference. In this paper we have formalized the connection between our technique and horn clause learning, showing that our use of vector space representations reduces to introducing disjunctions in the place of predicates in horn clause rules. The success of these simple techniques suggests that similar modifications might be fruitfully pursued in other logical inference systems. We have also shown preliminary work on extending random walk inference to a new domain: multiple choice question answering. With a naive approach to mapping the question answering problem to PRA, we have shown performance that is on par with state-of-the-art techniques on a small data set, and we have given some discussion of the strengths and weaknesses of this approach, including some promising directions for future research.

References

Carlson, A.; Betteridge, J.; Kisiel, B.; Settles, B.; Hruschka Jr, E. R.; and Mitchell, T. M. 2010. Toward an

architecture for never-ending language learning. In *AAAI*.

Clark, P.; Harrison, P.; and Balasubramanian, N. 2013. A study of the knowledge base requirements for passing an elementary science test. In *Proceedings of the 2013 workshop on Automated knowledge base construction*, 37–42. ACM.

Gardner, M.; Talukdar, P. P.; Kisiel, B.; and Mitchell, T. 2013. Improving learning and inference in a large knowledge-base using latent syntactic cues. In *Proceedings of EMNLP*. Association for Computational Linguistics.

Gardner, M.; Talukdar, P.; Krishnamurthy, J.; and Mitchell, T. 2014. Incorporating vector space similarity in random walk inference over knowledge bases. In *Proceedings of EMNLP*. Association for Computational Linguistics.

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

Lao, N.; Mitchell, T.; and Cohen, W. W. 2011. Random walk inference and learning in a large scale knowledge base. In *Proceedings of EMNLP*. Association for Computational Linguistics.

Quinlan, J. R. 1990. Learning logical definitions from relations. *Machine learning* 5(3):239–266.

Schoenmackers, S.; Etzioni, O.; Weld, D. S.; and Davis, J. 2010. Learning first-order horn clauses from web text. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, 1088–1098. Stroudsburg, PA, USA: Association for Computational Linguistics.

Schoenmackers, S.; Etzioni, O.; and Weld, D. S. 2008. Scaling textual inference to the web. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 79–88. Association for Computational Linguistics.

Talukdar, P. P.; Wijaya, D.; and Mitchell, T. 2012. Acquiring temporal constraints between relations. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, 992–1001. ACM.

Wang, W. Y.; Mazaitis, K.; and Cohen, W. W. 2013. Programming with personalized pagerank: A locally groundable first-order probabilistic logic. In *Proceedings of the 22Nd ACM International Conference on Conference on Information & Knowledge Management*, CIKM '13, 2129–2138. New York, NY, USA: ACM.