Large-Scale Knowledge Graph Identification using PSL

Extended Abstract

Jay Pujara and **Hui Miao** and **Lise Getoor**
Computer Science Dept.
University of Maryland
College Park, MD 20742
{jay,hui,getoor}@cs.umd.edu

William W. Cohen
Machine Learning Dept.
Carnegie Mellon University
Pittsburgh, PA 15213
wcohen@cs.cmu.edu

Introduction

The web is a vast repository of knowledge, but automatically extracting that knowledge, at scale, has proven to be a formidable challenge. A number of recent evaluation efforts have focused on automatic knowledge base population (Ji, Grishman, and Dang 2011; Artiles and Mayfield 2012), and many well-known broad domain and open information extraction systems exist, including the Never-Ending Language Learning (NELL) project (Carlson et al. 2010), OpenIE (Etzioni et al. 2008), and efforts at Google (Pasca et al. 2006), which use a variety of techniques to extract new knowledge, in the form of facts, from the web. These facts are interrelated, and hence, recently this extracted knowledge has been referred to as a *knowledge graph* (Singhal 2012). Unfortunately, most web-scale extraction systems do not take advantage of the rich dependencies found in the knowledge graph; instead approaches consider extractions independently, relying on simple heuristics to enforce consistency.

Recent work demonstrates that reasoning jointly is a promising approach to improving the knowledge graph. (Jiang, Lowd, and Dou 2012) choose candidate facts for inclusion in a knowledge base with a joint approach using Markov Logic Networks (MLNs) (Richardson and Domingos 2006). Jiang et al. provide a straightforward codification of ontological relations and candidate facts found in a knowledge base as rules in first-order logic and use MLNs to formulate a probabilistic model. However, due to the combinatorial explosion of Boolean assignments to random variables, inference and learning in MLNs pose intractable optimization problems. Jiang et al. limit the candidate facts they consider, restricting their dataset to a 2-hop neighborhood around each fact, and use a sampling approach to inference, estimating marginals using MC-SAT. Despite these approximations, their work demonstrate the utility of joint reasoning in comparison to a baseline that considers each fact independently.

Our work builds on the foundation of Jiang, Lowd, and Dou by providing a richer model for knowledge bases and vastly improving scalability. Our method transforms the noisy output of an information extraction system, a we define the problem of jointly inferring the entities, relations and attributes comprising a knowledge graph as *knowledge graph identification*. We leverage dependencies in the knowledge graph expressed through ontological constraints, and perform entity resolution allowing us to reason about co-referent entities. We also take advantage of uncertainty found in the extracted data, using continuous variables with values derived from extractor confidence scores.

To support this representation, we use a continuous-valued Markov random field and use the probabilistic soft logic (PSL) modeling framework (Broecheler, Mihalkova, and Getoor 2010). Inference in our model can be formulated as a convex optimization that scales linearly in the number of instances (Bach et al. 2012), allowing us to handle millions of candidate facts. These scalability gains allow us to evaluate our model on a predefined test set using millions of extractions in just 10 seconds, while also supporting lazy inference to produce a broader set of 4M candidates in just 2 hours. In this extended abstract we summarize contributions from a longer paper presented at ISWC 2013 (Pujara et al. 2013). Our work:

- Defines the knowledge graph identification problem;
- Uses soft-logic values to leverage extractor confidences;
- Formulates knowledge graph inference as convex optimization;
- Evaluates our proposed approach on extractions from NELL, a large-scale operational knowledge extraction system;

Knowledge Graph Identification

Our approach to constructing a consistent knowledge base uses PSL to represent the candidate facts from an information extraction system as a *knowledge graph* where entities are nodes, categories are labels associated with each node, and relations are directed edges between the nodes. Information extraction systems can extract such candidate facts, and these extractions can be used to construct a graph. Unfortunately, the output from an information extraction system is often incorrect; the graph constructed from it has spurious and missing nodes and edges, and missing or inaccurate node labels. Our approach to solving this problem builds on *graph identification* (Namata, Kok, and Getoor 2011), which identifies three key tasks: collective classification, link pre-

diction, and entity resolution. *Knowledge graph identification* provides a method for performing these three tasks in the presence of rich ontological information and multiple sources of uncertain information.

Unlike earlier work on graph identification, we use a very different probabilistic framework, PSL, allowing us to incorporate extractor confidence values and also support a rich collection of ontological constraints. PSL models are expressed using a set of universally-quantified logical rules which, when combined with the noisy extractions and ontological information, define a probability distribution over possible knowledge graphs. We explain how components of knowledge graph identification map to PSL rules, motivating these rules with examples of challenges found in a real-world information extraction system, the Never-Ending Language Learner (NELL) (Carlson et al. 2010).

Representation of Uncertain Extractions

NELL produces candidate extractions from a web corpus, which often contains noise. For example, candidate extractions from the NELL corpus include labels such as `bird(kyrgyzstan)` and `country(kyrghyzstan)` as well as relations such as `locatedIn(kyrghyzstan, Russia)`, `locatedIn(kyrgz republic, Asia)`, `locatedIn(kyrghyzstan, US)`, and `locatedIn(kyrgyzstan, Kazakhstan)`. These extractions can contain many mistakes that include variations in spelling and inconsistencies in relationships; clearly some of these candidate extractions are true while others are false, and NELL assigns confidence values to each extraction.

In PSL, we represent these candidate extractions with predicates CANDLBL and CANDREL , eg. CANDLBL(`kyrgyzstan`, `bird`) and CANDREL(`kyrgz republic`, `Asia`, `locatedIn`). In fact, NELL has multiple extractors that generate candidates, and we can use different predicates for each extractor. For a given extractor T, we introduce predicates CANDREL_T and CANDLBL_T to represent the candidates extracted by T. We relate these candidates to the unknown facts that we wish to infer, LBL and REL using the following rules:

$$\text{CANDREL}_T(E_1, E_2, R) \overset{w_{CR-T}}{\Rightarrow} \text{REL}(E_1, E_2, R)$$

$$\text{CANDLBL}_T(E, L) \overset{w_{CL-T}}{\Rightarrow} \text{LBL}(E, L)$$

Since PSL uses *soft logic*, we can represent noisy extractions by translating confidences into real-valued truth assignments in the $[0, 1]$ range. For example, if NELL extracts the relation `locatedIn(kyrgz republic, Asia)` and assigns it a confidence value of .9, we would assign the predicate CANDREL(`kyrgyzstan`, `Asia`, `locatedIn`) a soft-truth value of .9. Similarly, our output values for unknown facts are in the $[0, 1]$ range allow us to trade-off precision and recall by using a truth threshold. By learning the weights of these rules, w_{CL_T} and w_{CR_T}, our model combines multiple sources of information to label nodes and predict links.

Reasoning About Co-Referent Entities

While the previous PSL rules provide the building blocks of predicting links and labels using uncertain information, knowledge graph identification employs entity resolution to pool information across co-referent entities. In the example above, many different variant forms for the country Kyrgystan appear: `kyrgyzstan` , `kyrghyzstan` , and `kyrgz republic`. A key component of this process is identifying possibly co-referent entities and determining the similarity of these entities. We use the SAMEENT predicate to capture the similarity of two entities. While any similarity metric can be used, we compute the similarity of entities using a process of mapping each entity to the YAGO knowledge base(Suchanek, Kasneci, and Weikum 2007), extracting a set of Wikipedia articles for each entity and then computing the Jaccard index of possibly co-referent entities. We incorporate this information about co-referent entities by constraining the labels and relations of these entities through PSL rules:

$$\text{SAMEENT}(E_1, E_2) \tilde{\wedge} \text{LBL}(E_1, L) \overset{w_{EL}}{\Rightarrow} \text{LBL}(E_2, L)$$

$$\text{SAMEENT}(E_1, E_2) \tilde{\wedge} \text{REL}(E_1, E, R) \overset{w_{ER}}{\Rightarrow} \text{REL}(E_2, E, R)$$

$$\text{SAMEENT}(E_1, E_2) \tilde{\wedge} \text{REL}(E, E_1, R) \overset{w_{ER}}{\Rightarrow} \text{REL}(E, E_2, R)$$

These rules define an equivalence class of entities, such that all entities related by the SAMEENT predicate must have the same labels and relations. The soft-truth value of the SAMEENT, derived from our similarity function, mediates the strength of these rules. When two entities are very similar, they will have a high truth value for SAMEENT, so any label assigned to the first entity will also be assigned to the second entity. On the other hand, if the similarity score for two entities is low, the truth values of their respective labels and relations will not be strongly constrained.

Incorporating Ontological Information

Although entity resolution allows us to relate extractions that refer to the same entity, knowledge graphs can employ ontological information to specify rich relationships between many facts. Our ontological constraints are based on the logical formulation proposed in (Jiang, Lowd, and Dou 2012). Each type of ontological relation is represented as a predicate, and these predicates represent ontological knowledge of the relationships between labels and relations. For example, the constraints DOM(`hasCapital`, `country`) and RNG(`hasCapital`, `city`) specify that the relation `hasCapital` is a mapping from entities with label `country` to entities with label `city`. The constraint MUT(`country`, `bird`) specifies that the labels `country` and `bird`; are mutually exclusive, so that an entity cannot have both the labels `country` and `bird`. We similarly use constraints for subsumption of labels (SUB) and inversely-related functions (INV). To use this ontological knowledge, we introduce rules relating each ontological relation to the predicates representing our knowledge graph. We specify

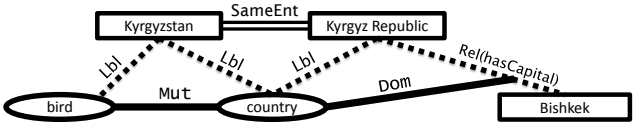

Figure 1: An illustration of the example showing how knowledge graph identification can resolve conflicting information in a knowledge graph. Entities are shown in rectangles, dotted lines represent uncertain information, solid lines show ontological constraints and double lines represent co-referent entities found with entity resolution.

seven types of ontological constraints in our experiments:

$$\textsc{Dom}(R, L) \quad \tilde{\wedge} \quad \textsc{Rel}(E_1, E_2, R) \quad \overset{w_Q}{\Rightarrow} \quad \textsc{Lbl}(E_1, L)$$

$$\textsc{Rng}(R, L) \quad \tilde{\wedge} \quad \textsc{Rel}(E_1, E_2, R) \quad \overset{w_Q}{\Rightarrow} \quad \textsc{Lbl}(E_2, L)$$

$$\textsc{Inv}(R, S) \quad \tilde{\wedge} \quad \textsc{Rel}(E_1, E_2, R) \quad \overset{w_Q}{\Rightarrow} \quad \textsc{Rel}(E_2, E_1, S)$$

$$\textsc{Sub}(L, P) \quad \tilde{\wedge} \quad \textsc{Lbl}(E, L) \quad \overset{w_Q}{\Rightarrow} \quad \textsc{Lbl}(E, P)$$

$$\textsc{RSub}(R, S) \quad \tilde{\wedge} \quad \textsc{Rel}(E_1, E_2, R) \quad \overset{w_Q}{\Rightarrow} \quad \textsc{Rel}(E_1, E_2, S)$$

$$\textsc{Mut}(L_1, L_2) \quad \tilde{\wedge} \quad \textsc{Lbl}(E, L_1) \quad \overset{w_Q}{\Rightarrow} \quad \tilde{\neg}\textsc{Lbl}(E, L_2)$$

$$\textsc{RMut}(R, S) \quad \tilde{\wedge} \quad \textsc{Rel}(E_1, E_2, R) \quad \overset{w_Q}{\Rightarrow} \quad \tilde{\neg}\textsc{Rel}(E_1, E_2, S)$$

Putting It All Together

Refining a knowledge graph becomes challenging as we consider the many interactions between the uncertain extractions that we encounter in knowledge graph identification (Figure 1). For example, NELL's ontology includes the constraint that the attributes `bird` and `country` are mutually exclusive. While extractor confidences may not be able to resolve which of these two labels is more likely to apply to `kyrgyzstan`, reasoning collectively using entity resolution and ontological constraints can provide a solution. For example, NELL is highly confident that `kyrgz republic` has a capital city, `Bishkek`. The NELL ontology specifies that the domain of the relation `hasCapital` has label `country`. Entity resolution allows us to infer that `kyrgz republic` refers to the same entity as `kyrgyzstan`. Deciding whether Kyrgyzstan is a bird or a country now involves a prediction where we include the confidence values of the corresponding `bird` and `country` facts from co-referent entities, as well as collective features from ontological constraints of these co-referent entities, such as the confidence values of the `hasCapital` relations.

To accomplish this collective reasoning, we use PSL to define a joint probability distribution over knowledge graphs. The universally-quantified rules described are a PSL model and provide the basis for defining this probability distribution. In a PSL program, $\mathbf{\Pi}$, this model is *grounded* by substituting values from NELL's noisy extractions into the rule template. For example, the rule:

$$\textsc{Dom}(R, L) \quad \tilde{\wedge} \quad \textsc{Rel}(E_1, E_2, R) \quad \overset{w_Q}{\Rightarrow} \quad \textsc{Lbl}(E_1, L)$$

can be grounded by substituting atoms from NELL, DOM(`hasCapital`, `country`), REL(`kyrgyzstan`, `Bishkek`, `hasCapital`), and LBL(`kyrgyzstan`, `country`), into the rule template. We refer to the collection of ground rules in the program as R.

Unlike Boolean logic where each grounding would have a binary truth value, our choice of soft-logic requires a different definition of truth value. We term an assignment of soft-truth values to atoms an interpretation, I, and use the *Lukasiewicz t-norm* and *co-norm* to determine the truth values of logical formulas under an interpretation. This t-norm defines a relaxation of the logical connectives (denoted using $\sim$) AND ($\wedge$), OR($\vee$), and NOT($\neg$), as follows:

$$p\tilde{\wedge}q = \max(0, p + q - 1)$$
$$p\tilde{\vee}q = \min(1, p + q)$$
$$\tilde{\neg}p = 1 - p$$

With this definition, we can assign a truth value $T_r(I)$ to each grounding $r \in R$ and define a *distance to satisfaction*, $\phi_r(I) = 1 - T_r(I)$ for each grounding. The probability distribution over knowledge graphs, $P_{\mathbf{\Pi}}(G)$ can now be defined in terms of the probability of an interpretation, $f(I)$, using a weighted combination of the distance to satisfaction of ground rules in the PSL program:

$$P_{\mathbf{\Pi}}(G) = f(I) = \frac{1}{Z}\exp\left[-\sum_{r \in R} w_r \phi_r(I)^p\right]$$

where $p \in 1, 2$ specifies a linear or quadratic combination and Z is a normalization constant.

Most Probable Explanation (MPE) inference corresponds to finding an interpretation that maximizes $f(I)$, which can then be mapped to a set of labels and relations that comprise the true knowledge graph. In our work, we choose a soft-truth threshold and determine the true entities, labels and relations by using those atoms whose truth value exceeds the threshold. MPE inference can be formulated as convex optimization in PSL, and using the Alternating Direction Method of Multipliers (ADMM), (Bach et al. 2012) have shown performance that scales linearly with the number of ground rules in the PSL program.

Table 1: Comparing against previous work on the NELL dataset, knowledge graph identification using PSL demonstrates a substantive improvement.

Method	AUC	Prec	Recall	F1
Baseline	0.873	0.781	0.881	0.828
NELL	0.765	0.801	0.580	0.673
MLN	0.899	**0.837**	0.837	0.836
PSL-KGI	**0.904**	0.777	**0.944**	**0.853**

Experimental Results

We compare our method to data from iteration 165 of NELL using previously reported results on a manually-labeled evaluation set (Jiang, Lowd, and Dou 2012). The dataset contains 1.7M candidate facts, 440K previously promoted facts, and nearly 80K ontological relationships. A summary of our results on this dataset is shown in Table 1.

The first method we compare to is a simple baseline where candidates are given a soft-truth value equal to the extractor

confidence (averaged across extractors when appropriate). Results are reported at a soft-truth threshold of .45 which maximizes F1. We also compare the default strategy used by the NELL project to choose candidate facts to include in the knowledge base. Their method uses the ontology to check the consistency of each proposed candidate with previously promoted facts already in the knowledge base. Candidates that do not contradict previous knowledge are ranked using a heuristic rule, and the top candidates are chosen for promotion subject to score and rank thresholds. Note that the NELL method includes judgments for all input facts, not just those in the test set.

The third method we compare against is the best-performing MLN model from (Jiang, Lowd, and Dou 2012) which expresses ontological constraints, and candidate and promoted facts through logical rules similar to those in our model. The MLN uses additional predicates that have confidence values taken from a logistic regression classifier trained using manually labeled data. The MLN uses hard ontological constraints, learns rule weights considering rules independently and using logistic regression, scales weights by the extractor confidences, and uses MC-Sat with a restricted set of atoms to perform approximate inference, reporting output at a .5 marginal probability cutoff, which maximizes the F1 score. The MLN method only generates predictions for a 2-hop neighborhood generated by conditioning on the values of the query set.

Our method, PSL-KGI, uses PSL with quadratic, weighted rules for ontological constraints, entity resolution, and candidate and promoted facts as well as incorporating a prior. We also incorporate the predicates generated for the MLN method for a more equal comparison. We learn weights for all rules, including the prior, using a voted perceptron learning method. The weight learning method generates a set of target values by running inference and conditioning on the training data, and then chooses weights that maximize the agreement with these targets in absence of training data. Since we represent extractor confidence values as soft-truth values, we do not scale the weights of these rules. Using the learned weights, we perform inference on the same neighborhood defined by the query set that is used by the MLN method. We report these results, using a soft-truth threshold of .55 to maximize F1, as PSL-KGI.

As Table 1 shows, knowledge graph identification produces modest improvements in both F1 and AUC. Adding entity resolution and source information to our model while using soft-truth values for facts provide a richer representation of the data. One motivation for using PSL for knowledge graph identification was to frame complex joint reasoning as a convex optimization. This model can tackle problems such as the NELL dataset, containing millions of candidate facts. Running inference for knowledge graph identification given a query set (PSL-KGI) requires a mere 10 seconds to perform inference. The MLN method we compare against takes a few minutes to an hour to run. Running knowledge graph identification to produce the complete knowledge graph containing 4.9M facts requires only 130 minutes, while such problems are infeasible using existing MLN optimization techniques.

Conclusion

Knowledge graphs present a Big Data problem: reasoning collectively about millions of interrelatd facts. We formulate the problem of *knowledge graph identification*, jointly inferring a knowledge graph from the noisy output of an information extraction system through a combined process of determining co-referent entities, predicting relational links, collectively classifying entity labels, and enforcing ontological constraints. Using PSL, we illustrate the scalability benefits of our approach on a large-scale dataset from NELL, while producing high-precision results. Our method provides a substantial increase in F1 score while also improving AUC and scales linearly with the number of ground rules. In practice, we show that on a NELL dataset our method can infer a full knowledge graph in just two hours or make predictions on a known query set in a matter of seconds.

Acknowledgments This work was partially supported by NSF CAREER grant 0746930 and NSF grants IIS1218488 and CCF0937094. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

Artiles, J., and Mayfield, J., eds. 2012. *Workshop on Knowledge Base Population*.

Bach, S. H.; Broecheler, M.; Getoor, L.; and O'Leary, D. P. 2012. Scaling MPE Inference for Constrained Continuous Markov Random Fields with Consensus Optimization. In *NIPS*.

Broecheler, M.; Mihalkova, L.; and Getoor, L. 2010. Probabilistic Similarity Logic. In *UAI*.

Carlson, A.; Betteridge, J.; Kisiel, B.; Settles, B.; Hruschka, E. R.; and Mitchell, T. M. 2010. Toward an Architecture for Never-Ending Language Learning. In *AAAI*.

Etzioni, O.; Banko, M.; Soderland, S.; and Weld, D. S. 2008. Open Information Extraction from the Web. *Communications of the ACM* 51(12).

Ji, H.; Grishman, R.; and Dang, H. 2011. Overview of the Knowledge Base Population Track. In *Text Analysis Conference*.

Jiang, S.; Lowd, D.; and Dou, D. 2012. Learning to Refine an Automatically Extracted Knowledge Base Using Markov Logic. In *ICDM*.

Namata, G. M.; Kok, S.; and Getoor, L. 2011. Collective Graph Identification. In *KDD*.

Pasca, M.; Lin, D.; Bigham, J.; Lifchits, A.; and Jain, A. 2006. Organizing and Searching the World Wide Web of Facts-Step One: the One-million Fact Extraction Challenge. In *AAAI*.

Pujara, J.; Miao, H.; Getoor, L.; and Cohen, W. 2013. Knowledge graph identification. In *ISWC*.

Richardson, M., and Domingos, P. 2006. Markov Logic Networks. *Machine Learning* 62(1-2).

Singhal, A. 2012. Introducing the Knowledge Graph: Things, Not Strings. Official Blog (of Google).

Suchanek, F. M.; Kasneci, G.; and Weikum, G. 2007. Yago: A Core of Semantic Knowledge. In *WWW*.