Using Semantics and Statistics to Turn Data into Knowledge

Jay Pujara, Hui Miao, Lise Getoor, William W. Cohen

■ *Many information-extraction and knowledge base construction systems are addressing the challenge of deriving knowledge from text. A key problem in constructing these knowledge bases from sources like the web is overcoming the erroneous and incomplete information found in millions of candidate extractions. To solve this problem, we turn to semantics — using ontological constraints between candidate facts to eliminate errors. In this article, we represent the desired knowledge base as a knowledge graph and introduce the problem of knowledge graph identification, collectively resolving the entities, labels, and relations present in the knowledge graph. Knowledge graph identification requires reasoning jointly over millions of extractions simultaneously, posing a scalability challenge to many approaches. We use probabilistic soft logic (PSL), a recently introduced statistical relational learning framework, to implement an efficient solution to knowledge graph identification and present state-of-the-art results for knowledge graph construction while performing an order of magnitude faster than competing methods.*

A growing body of research focuses on extracting knowledge from text such as news reports, encyclopedic articles, and scholarly research in specialized domains. Much of this data is freely available on the World Wide Web and harnessing the knowledge contained in millions of web documents remains a problem of particular interest. The scale and diversity of this content pose a formidable challenge for systems designed to extract this knowledge. Many well-known broad domain and open information-extraction systems seek to build knowledge bases from text, including the Never-Ending Language Learning (NELL) project (Carlson et al. 2010), OpenIE (Etzioni et al. 2008), DeepDive (Niu et al. 2012), and efforts at Google (Pasca et al. 2006). Ultimately, these information-extraction systems produce a collection of candidate facts that include a set of entities, attributes of these entities, and the relations between these entities.

Information-extraction systems use a sophisticated collection of strategies to generate candidate facts from web documents, spanning the syntactic, lexical, and structural features of text (Weikum and Theobald 2010, Wimalasuriya and Dou 2010). Although these systems are capable of extracting many candidate facts from the web, their output is often hampered by noise. Documents contain inaccurate, outdated, incomplete, or hypothetical information, and informal and creative language used in web documents is often difficult to interpret. As a result, the candidates produced by information-extraction systems often miss key facts and include spurious outputs, compromising the usefulness of the extractions. To combat such noise, information-extraction systems capture a vast array of features and statistics, ranging from the characteristics of the web pages used to generate extractions to the reliability of the particular patterns or techniques used to extract information. Using this host of features and a modest amount of training data, many information-extraction systems employ heuristics or learned prediction functions to assign a confidence score to each candidate fact. These confidence scores capture the inherent uncertainty in the text from which the facts were extracted, and can ideally be used to improve the quality of the knowledge base.

Although many information-extraction systems use features derived from text to measure the quality of candidate facts, few take advantage of the many semantic dependencies between these facts. For example, many categories, such as "male" and "female" may be mutually exclusive, or restricted to a subset of entities, such as living organisms. Recently, the semantic web movement has developed standards and tools to express these dependencies through ontologies designed to capture the diverse information present on the Internet. The problem of building domain-specific ontologies for expert users with semantic web tools is challenging and well researched, with high-quality ontologies for domains including bioinformatics, media such as music and books, and governmental data. More general ontologies have been developed for broad collections such as the online encyclopedia Wikipedia. These semantic constraints are valuable for improving the quality of knowledge bases, but incorporating these dependencies into existing information-extraction systems is not straightforward.

The constraints imposed by an ontology are generally constraints between facts. For example, candidate facts assigning a particular entity to the categories "male," "female," and "living organism" are interrelated. Hence, leveraging the dependencies between facts in a knowledge base requires reasoning jointly about the extracted candidates. Due to the large scale at which information-extraction systems operate, considering the dependencies between millions of candidates presents a scalability challenge. Logic-based approaches to reasoning about these candidates, such as automated theorem proving or constraint processing systems, are often impractical due to the uncertainty and errors in the candidate facts (Hitzler and van Harmelen 2010). We describe a statistical relational learning (Getoor and Taskar 2007) approach to the problem of constructing knowledge bases, applying recently developed techniques that incorporate statistical information and logical dependencies at scale.

Our approach uses a lightweight representation that we refer to as a *knowledge graph.* The knowledge graph contains the facts in the knowledge base, similar to the ABox in traditional knowledge representation approaches. In the knowledge graph, entities are represented as nodes, the attributes of each entity are node labels, and relationships between two or more entities are represented as edges. To build the knowledge graph, we use as input the entities, labels, and relationships produced by an information-extraction system, which can be represented as an extraction graph. Noise, ambiguity, and errors limit the usefulness of the extraction graph. In this article, we summarize work on knowledge graph identification (Pujara et al. 2013), a process that infers a knowledge graph from the extraction graph. In addition to the statistical properties of the extraction graph, knowledge graph identification incorporates semantics, in the form of an ontology and ontological constraints defined over the facts in the knowledge graph, to leverage dependencies between facts. These ontological constraints correspond to the TBox in knowledge representation terms. Figure 1 illustrates the noisy extraction graph that serves as an input to knowledge graph identification and the knowledge graph produced as output.

Statistical relational learning approaches can capture both the structure of the knowledge graph as well as the logical dependencies between the constituent facts. Unlike traditional reasoning systems, statistical relational learning approaches can treat ontological constraints as weighted rules, using them as "hints" to find the correct facts in a knowledge graph. For example, Jiang, Lowd, and Dou (2012) have demonstrated the effectiveness of Markov logic networks (MLNs) for jointly reasoning about the facts in a knowledge base.

Although MLNs provide a powerful approach to producing knowledge bases, their principal weakness is scalability. Here, instead, we use probabilistic soft logic (PSL) (Broecheler, Mihalkova, and Getoor 2010), a recently developed statistical relational learning approach. PSL shares many attractive features of MLNs: models can be specified using predicates and rules written in first-order logic syntax and translated into a probabilistic graphical model. In addition, PSL overcomes the scalability limitations of MLNs. Each logical predicate in an MLN has a Boolean truth value, and inference in these models is an intractable combinatorial optimization. As a result, many approximate methods for optimizing such models rely on sampling techniques for tractable inference. One of the key features of PSL is that predicates take continuous values in the [0, 1] range. By relaxing truth values to the continuous domain, PSL is able to solve inference tasks as an efficient convex optimization. In addition to improving

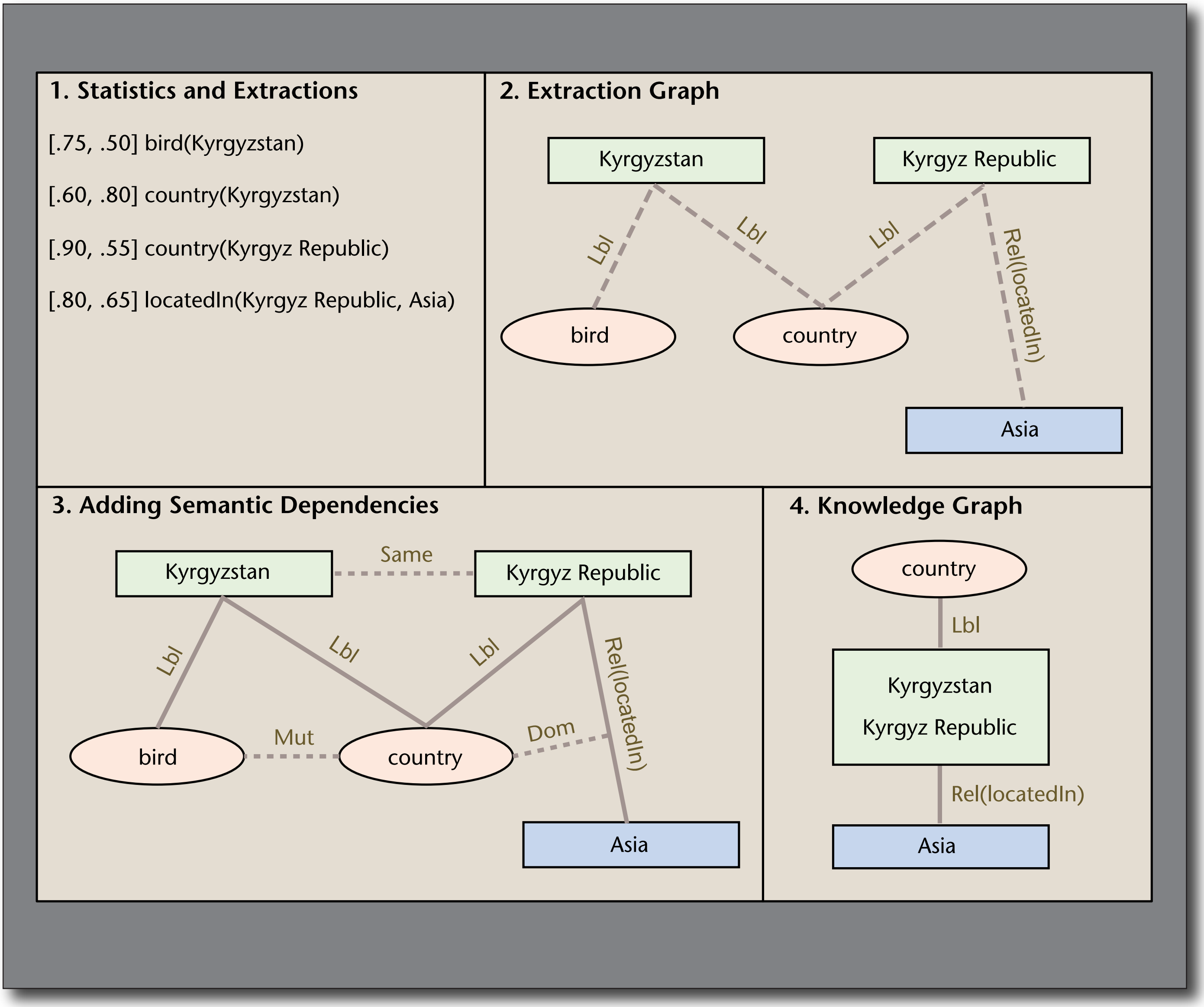

Figure 1. An Illustration of the Knowledge Graph Construction Process.

The illustration shows (1) the conflicting candidate facts and statistics produced by an information-extraction system, (2) their representation as an uncertain extraction graph, (3) semantic dependencies between facts introduced by an ontology, and (4) the final knowledge graph where errors have been removed by knowledge graph identification. Entities are shown in rectangles, labels are shown in circles, dashed lines represent uncertain information, dotted lines are used for ontological constraints and entity resolution, and solid lines represent the knowledge graph produced by knowledge graph identification.

scalability, continuous truth values provide a more suitable representation for statistical features, such as the confidence values of candidate facts. In this article, we show how implementing a PSL model for knowledge graph identification allows us to reason jointly over millions of facts efficiently, producing new state-of-the-art results. In addition, the improved scalability of our approach allows us to operate on data sets that are intractable for competing approaches to joint inference, such as MLNs.

Knowledge Graph Identification: The Problem

The problem of jointly inferring the entities, labels, and relations in a graph from uncertain data through the processes of entity resolution, collective classification, and link prediction is referred to as graph identification (Namata, Kok, and Getoor 2011). Similar to graph identification, knowledge graph identification completes these three tasks to infer the most probable knowledge graph from uncertain extractions. However, unlike graph identification, knowledge graph identification incorporates ontological constraints between facts during inference. We motivate the challenges presented by entity resolution, collective classification, and link prediction in knowledge graphs, then demonstrate how ontological information plays a vital role in each task. We draw examples from NELL, a large-scale information-extraction system with a rich ontology, but the problems we identify are widespread in information extraction.

Entity extraction has a common problem: many textual references that initially look different may refer to the same real-world entity. For example, NELL's knowledge base contains candidate facts involving the entities "kyrghyzstan," "kyrgzstan," "kyrgystan," "kyrgyz republic," "kyrgyzstan," and "kyrgistan," which are all variants or misspellings of the country Kyrgyzstan. In the extraction graph, each of these entities incorrectly corresponds to different nodes. Our approach uses entity resolution to determine coreferent entities in the knowledge graph, producing a consistent set of labels and relations for each resolved node.

Another challenge in knowledge graph construction is inferring labels consistently. For example, NELL's extractions assign Kyrgyzstan the labels "country" as well as "bird." Ontological information suggests that an entity is very unlikely to be both a country and a bird simultaneously. Moreover, other extractions, for example that Bishkek is the capital of Kyrgyzstan, support the conclusion that Kyrgyzstan is a country. Using the labels of related entities in the knowledge graph can allow us to determine the correct label of an entity. Our approach uses collective classification to label nodes in a manner that takes into account ontological information and neighboring labels.

A third problem commonly encountered in knowledge graphs is determining the relationships between entities. NELL also has many facts relating the location of Kyrgyzstan to other entities. These candidate relations include statements that Kyrgyzstan is located in Kazakhstan, Kyrgyzstan is located in Russia, Kyrgyzstan is located in the former Soviet Union, Kyrgyzstan is located in Asia, and that Kyrgyzstan is located in the United States. Some of these possible relations are true, while others are clearly false and contradictory. Our approach uses link prediction to predict edges in a manner that takes into account ontological information and the rest of the inferred structure.

Each of these tasks is extremely challenging when posed as straightforward prediction tasks that do not use dependencies, and using only features from the information-extraction system can cause these tasks to be underconstrained. Incorporating ontological constraints often resolves such difficulties. For example, understanding that coreferent entities such as "kyrgyzstan" and "kyrgyz republic" are the same entity and, as a result, should have the same labels and relations improves the quality of predicted labels and relations while also resolving ambiguity about these entities.

Multiple ontological constraints work in concert to improve the knowledge graph. NELL's ontology includes the constraint that the labels "bird" and "country" are mutually exclusive. Ontological constraints also require the "locatedIn" relation to be a mapping from a domain of countries to a range of continents. Combining these constraints in knowledge graph identification produces dependencies between the location of entities and their potential labels. For example, the input to knowledge graph identification includes the erroneous extraction stating that "kyrgyzstan" has label "bird." By combining the extractions that "kyrgyz republic" is located in Asia, and hence has label "country," that "kyrgyz republic" and "kyrgyzstan" are coreferent, and that "bird" and "country" are mutually exclusive, we are able to remove the erroneous "bird" label. This complex set of dependencies requires jointly reasoning about millions of extractions simultaneously — a challenge we address in our model for knowledge graph identification.

Knowledge Graph Identification: The Solution

Knowledge graph identification requires combining two disparate elements: the statistics output by an information extraction and ontological constraints derived from the semantics of the knowledge graph. In this section, we describe a model for knowledge graph identification using rules written in first-order logic syntax. We define predicates that capture candidate extractions, coreference information, and ontological knowledge and introduce rules that capture the relationships between these elements and the facts contained in the knowledge graph. However, since the inputs to this model are uncertain values and statistics from an extraction system, the logical atoms in the rules take values between 0 and 1. We combine the rules and statistical inputs in a probabilistic graphical model to determine which facts to include in the knowledge graph.

In order to implement knowledge graph identification, our model uses probabilistic soft logic (Broecheler, Mihalkova, and Getoor 2010), a recently introduced framework for specifying probabilistic graphical models over continuously valued random variables. PSL provides many advantages: models are easily defined using declarative rules with first-order logic syntax, continuously valued variables provide a convenient representation of uncertainty, weighted rules and weight learning capture the importance of model rules, and advanced features such as set-based aggregates and hard constraints are supported. Using PSL, we can transform the statistics and rules in our knowledge graph identification model into a probability distribution over knowledge graphs, and then infer the most likely knowledge graph. A significant obstacle to knowledge graph construction is scale — reasoning jointly over the millions of facts found in a knowledge graph is intractable in many models. However, in PSL this joint optimization is formulated as a convex objective that is highly scalable allowing us to handle millions of facts in minutes. After introducing each set of rules in our knowledge graph identification model, we will show how PSL transforms these rules into a probability distribution over knowledge graphs.

Representation of Uncertain Extractions

Information-extraction systems use a collection of techniques that operate on document features such as the structural elements (for example, tables) lexical patterns (for example, the phrase "president of"), or morphological features (for example, capitalization). Each extractor produces a different set of outputs, and may assign each output a confidence value. The first step of building a knowledge graph is combining features and extractions from different extractors.

For example, an extractor based on structural elements might produce the label *bird(Kyrgyzstan)* and the relation *locatedIn(Kyrgyz Republic, Asia)* while a pattern-based classifier might produce the label *country(Kyrghyzstan)* as well as relations such as *hasCapital(Kyrgyz Republic, Bishkek)*. We use a different predicate for the candidates generated by each extractor. For a given extractor *T,* we introduce predicates $\textsc{CandRel}_T$ and $\textsc{CandLbl}_T$ 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:

$$\mathbf{w}_{\text{CR-}T} : \textsc{CandRel}_T\,(E_1, E_2, R) \Rightarrow \textsc{Rel}\,(E_1, E_2, R)$$
$$\mathbf{w}_{\text{CL-}T} : \textsc{CandLbl}_T\,(E, L) \Rightarrow \textsc{Lbl}\,(E, L)$$

We define weights $\mathbf{w}_{\text{CR-}T}$ and $\mathbf{w}_{\text{CL-}T}$ for the relations and labels produced by extractor *T,* allowing us to compensate for the differing reliability of each technique. Using training data, we can use PSL to learn these weights. As a concrete example, a grounding of the structural extractor's candidate *bird(Kyrgyzstan)* would produce the formula:

$$\textsc{CandLbl}_{\text{struct}}(\mathit{Kyrgyzstan, bird}) \Rightarrow \textsc{Lbl}(\mathit{Kyrgyzstan, bird})$$

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 the label extraction *bird(Kyrgyzstan)* has a confidence value of 0.6, we would assign the predicate $\textsc{CandLbl}_{\text{struct}}(\mathit{Kyrgyzstan, bird})$ a soft-truth value of 0.6. While these simple rules associate uncertain inputs with the facts in the knowledge graph, more complex rules allow us to incorporate knowledge about coreferent entities.

Reasoning About Coreferent Entities

Entity resolution identifies potentially coreferent entities and assigns a similarity score for each pair of entities. In the example above, many different variant forms for the country Kyrgystan appear: *Kyrgyzstan, Kyrghyzstan,* and *Kyrgyz Republic*. Knowledge graph identification employs entity resolution to pool information across these coreferent entities. We introduce the following rules to constrain the labels and relations of these coreferent entities:

$$\mathbf{w}_{\mathbf{EL}} : \textsc{SameEnt}(E_1, E_2) \wedge \textsc{Lbl}\,(E_1, \mathrm{L}) \Rightarrow \textsc{Lbl}\,(E_2, L)$$
$$\mathbf{w}_{\mathbf{ER}} : \textsc{SameEnt}(E_1, E_2) \wedge \textsc{Rel}\,(E_1, \mathrm{E}, \mathrm{R}) \Rightarrow \textsc{Rel}\,(E_2, E, R)$$
$$\mathbf{w}_{\mathbf{ER}} : \textsc{SameEnt}(E_1, E_2) \wedge \textsc{Rel}\,(E, E_1, \mathrm{R}) \Rightarrow \textsc{Rel}\,(E, E_2, R)$$

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 by 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 domain and range constraints Dom*(locatedIn, country)* and Rng*(locatedIn, continent)* specify that the relation *locatedIn* is a mapping from entities with label *country* to entities with label *continent.* The mutual exclusion 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 seven types of ontological constraints in our experiments:

$$\mathbf{w}_{\mathbf{O}} : \textsc{Dom}(R, L) \wedge \textsc{Rel}\,(E_1, E_2, R) \Rightarrow \textsc{Lbl}\,(E_1, L)$$
$$\mathbf{w}_{\mathbf{O}} : \textsc{Rng}(R, L) \wedge \textsc{Rel}\,(E_1, E_2, R) \Rightarrow \textsc{Lbl}\,(E_2, L)$$
$$\mathbf{w}_{\mathbf{O}} : \textsc{Inv}\ R, S) \wedge \textsc{Rel}\,(E_1, E_2, R) \Rightarrow \textsc{Rel}\,(E_2, E_1, S)$$
$$\mathbf{w}_{\mathbf{O}} : \textsc{Sub}(L, P\,) \wedge \textsc{Lbl}\,(E, L) \Rightarrow \textsc{Lbl}\,(E, P\,)$$
$$\mathbf{w}_{\mathbf{O}} : \textsc{RSub}(R, S) \wedge \textsc{Rel}\,(E_1, E_2, R) \Rightarrow \textsc{Rel}\,(E_1, E_2, S)$$
$$\mathbf{w}_{\mathbf{O}} : \textsc{Mut}(L_1, L_2) \wedge \textsc{Lbl}\,(E, L_1) \Rightarrow \neg\,\textsc{Lbl}\,(E, L_2)$$
$$\mathbf{w}_{\mathbf{O}} : \textsc{RMut}(R, S) \wedge \textsc{Rel}\,(E_1, E_2, R) \Rightarrow \neg\,\textsc{Rel}\,(E_1, E_2, S)$$

$[\phi_1]$ $\textsc{CandLbl}_{\text{struct}}(\texttt{Kyrgyzstan}, \texttt{bird})$
$\quad\Rightarrow$ $\textsc{Lbl}(\texttt{Kyrgyzstan}, \texttt{bird})$
$[\phi_2]$ $\textsc{CandRel}_{\text{pat}}(\texttt{Kyrgyz Rep.}, \texttt{Asia}, \texttt{locatedIn})$
$\quad\Rightarrow$ $\textsc{Rel}(\texttt{Kyrgyz Rep.}, \texttt{Asia}, \texttt{locatedIn})$
$[\phi_3]$ $\textsc{SameEnt}(\texttt{Kyrgyz Rep.}, \texttt{Kyrgyzstan})$
$\quad\wedge$ $\textsc{Lbl}(\texttt{Kyrgyz Rep.}, \texttt{country})$
$\quad\Rightarrow$ $\textsc{Lbl}(\texttt{Kyrgyzstan}, \texttt{country})$
$[\phi_4]$ $\textsc{Dom}(\texttt{locatedIn}, \texttt{country})$
$\quad\wedge$ $\textsc{Rel}(\texttt{Kyrgyz Rep.}, \texttt{Asia}, \texttt{locatedIn})$
$\quad\Rightarrow$ $\textsc{Lbl}(\texttt{Kyrgyz Rep.}, \texttt{country})$
$[\phi_5]$ $\textsc{Mut}(\texttt{country}, \texttt{bird})$
$\quad\wedge$ $\textsc{Lbl}(\texttt{Kyrgyzstan}, \texttt{country})$
$\quad\Rightarrow$ $\neg\textsc{Lbl}(\texttt{Kyrgyzstan}, \texttt{bird})$

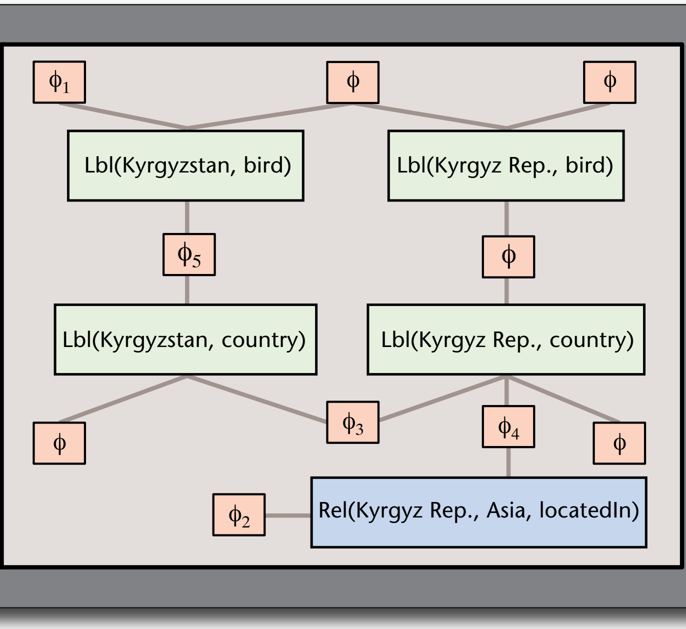

Figure 2. An Example of the Graphical Model Used in Knowledge Graph Identification.

This figure shows a subset of the probabilistic graphical model defined by the rules in PSL in our example. Each potential fact in the knowledge graph is a variable, and factors (ϕ) represent a distance to satisfaction capturing dependencies between variables.

Putting It All Together

Constructing a knowledge graph is challenging because of the many interactions between uncertain extractions, coreferences, ontological information, and facts in the knowledge graph. To capture the complex set of dependencies in the knowledge graph, we formulate the problem as a probabilistic graphical model. Figure 2 illustrates a small portion of the graphical model associated with the example in this section.

Each possible fact in the knowledge graph is a variable in the model. Dependencies between variables derived are from rules. These dependencies are shown as factors between variables, shown using ϕ labels. Each ϕ is a function of the variable values that measures a distance to satisfaction of the variables' truth values relative to the rules, where a high distance to satisfaction indicates a violated rule or constraint. For example, if the variables LBL*(Kyrgyzstan,bird)* and LBL*(Kyrgyzstan,country)* both had high truth values, the factor ϕ_5 representing a mutual exclusion constraint between these variables would have a high distance to satisfaction. Similarly, the factor ϕ_3 will have a low distance to satisfaction for coreferent entities *Kyrgyzstan* and *Kyrgyz Republic* when both have the same label, country.

To determine the value of each variable in the model, we use PSL to define a joint probability distribution over the different possible knowledge graphs. The universally quantified rules described above form a PSL model and provide the basis for defining this probability distribution. In a PSL program, Π, this model is grounded by substituting values from the extractions and ontology into the rule templates. 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 relax truth values in the knowledge graph G to the [0, 1] interval and use a logically consistent interpretation to determine the truth values of ground logical formulas. With this definition, we can assign a truth value $T_r(G)$ to each grounding $r \in R$ and define a distance to satisfaction, $\phi_r(G) = (1 - T_r(G))^2$ for each grounding. The probability distribution over knowledge graphs, $P_\Pi(G)$ can now be defined in terms of the weighted combination of the distances to satisfaction of ground rules in the PSL program:

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

Given this distribution, the task of most probable explanation (MPE) inference corresponds to finding the soft-truth values of every fact in the knowledge graph G that maximize the value of this probability distribution. The soft-truth values of variables can be interpreted as confidences. 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. By using a small validation set to choose this threshold, we can balance precision and recall of the resulting knowledge graph as required by a particular application.

The myriad dependencies in the knowledge graph identification model are a formidable scalability challenge. In PSL, MPE inference can be formulated as convex optimization. Solving this convex optimization 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. In practice, this allows our model implementing knowledge graph identification to jointly infer the values of millions of variables in the knowledge graph in just hours — a result we detail in the next section.

Evaluation

We highlight the effectiveness of knowledge graph identification with results for building a knowledge graph with data from NELL, a large-scale information-extraction system operating on text from the web. Our experiments contrast two very different evaluation settings. The first, used in prior work, restricts knowledge graph inference to a small subset of variables and excludes some contradictory values. Our results in this simpler setting improve on the state-of-the-art results while completing the inference task in just seconds. In the second evaluation setting, we apply knowledge graph identification to infer the complete knowledge graph, operating over the space of all possible knowledge graphs and handling millions of variables and tens of millions of dependencies. Despite the magnitude of this inference task, our implementation of knowledge graph identification requires just over two hours to complete and improves on the performance of the existing NELL system. Data and code for these experiments are available on GitHub.[1]

NELL Data Set

The Never-Ending Language Learner is a system that seeks to iteratively create a knowledge base by constantly improving its ability to process text and extract information. In each iteration, NELL uses facts learned from the previous iteration and a corpus of web pages to generate a new set of candidate facts. Our experimental results are on data from the 165th iteration of NELL, using the candidate facts, previously promoted facts, and ontological relationships that NELL used during that iteration. We summarize the important statistics of this data set in table 1. NELL uses diverse extraction techniques, and we use distinct predicates for the most prominent sources, while averaging values across extractors that do not contribute a significant number of facts. In addition to these candidate facts, NELL uses a heuristic formula to selectively promote candidates in each iteration of the system into a knowledge base, however these promotions are often noisy so the system assigns each promotion a confidence value. We represent these promoted candidates from previous iterations as an additional source with corresponding candidate predicates.

Cand. Label	1.2M
Cand. Rel	100K
Promotions	440K
Unique Labels	235
Unique Rels	221
Dom	418
Rng	418
Inv	418
Mut	17.4K
RMut	48.5K
Sub	288
RSub	461

Table 1. Summary of Data Set Statistics for NELL.

The table includes the number of candidate facts in input data, the distinct relations and labels present, and the number of ontological relationships defined between these relations and labels.

Entity Coreference

Knowledge graph identification also incorporates entity coreferences, a feature missing from the NELL data. We derive entity coreference by using the YAGO database (Suchanek, Kasneci, and Weikum 2007) as part of our entity resolution approach. The YAGO database contains entities that correspond to Wikipedia articles, variant spellings and abbreviations of these entities, and associated WordNet categories. We match entity names in NELL with YAGO entities. We perform selective stemming on the NELL entities, employ blocking on candidate labels, and use a case-insensitive string match to find corresponding YAGO entities. Once we find a matching set of YAGO entities, we can generate a set of Wikipedia URLs that map to the corresponding NELL entities. Our model uses a SameEnt predicate to capture the similarity of two entities. We then define a similarity function on the article URLs and use the computed similarity as the soft-truth value of the SameEnt predicate. For our similarity score we use the Jaccard index, the ratio of the size of the set intersection and the size of the set union.

Evaluation Scenarios

In our experiments using NELL, we consider two scenarios. The first is similar to the experimental setup in the paper by Jiang, Lowd, and Dou (2012) where rule weights are learned using training data and pre-

Method	AUC	F1
Baseline	0.873	0.828
NELL	0.765	0.673
MLN	0.899	0.836
PSL-KGI	**0.904**	**0.853**

Table 2. Comparing against Previous Work on the NELL Data set.

Knowledge graph identification using PSL demonstrates a substantive improvement. The best-performing method is shown in boldface.

Method	AUC	F1
PSL-NoSrcs	0.900	0.852
PSL-NoER	0.899	**0.853**
PSL-NoOnto	0.887	0.826
PSL-KGI	**0.904**	**0.853**

Table 3. Comparing Variants of PSL Graph Identification.

This comparison shows the importance of ontological information, but the best performance is achieved when all of the components of knowledge graph identification are combined.

dictions are made on a limited 2-hop neighborhood of the test set. The neighborhood used in this previous work attempts to improve scalability by generating a grounding of the rules using atoms in the test set and only including additional atoms that are not trivially satisfied in this grounding. In practice, this produces a neighborhood that is distorted by omitting atoms that may contradict those in the test set. For example, if ontological relationships such as Sub*(country,location)* and Mut*(country, city)* are present, the test set atom Lbl(*Kyrgyzstan,country*) would not introduce Lbl*(Kyrgyzstan,city)* or Lbl*(Kyrgyzstan, location)* into the neighborhood, even if contradictory data were present in the input candidates. By removing the ability to reason about contradictory information, we believe this evaluation setting diminishes the true difficulty of the problem. We validate our approach on this setting, but also present results from a more realistic setting. In the second scenario we perform inference independently of the test set, lazily generating target variables for atoms supported by the input data, using a soft-truth value threshold of .01. This second setting allows us to infer a complete knowledge graph with truth values for all possible variables, including those that may contradict the atoms in the test set.

Knowledge Graph Identification Results for NELL

We compare our method against previously reported results on a manually labeled evaluation set of 4500 facts (Jiang, Lowd, and Dou 2012). A summary of these results is shown in table 2, where the best-performing method is shown in boldface. The first method we compare to is a 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 based on the confidence scores of the extractors, 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 MCMC 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, as described earlier.

Our method, PSL-KGI, implements the KGI model in PSL using weighted rules for ontological constraints, entity resolution, and candidate and promoted facts as well as incorporating generic prior for every fact in the knowledge graph.. 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.

In table 2 we report area under the precision-recall curve (AUC) and F1 measure, the harmonic mean of the precision and recall. Our implementation of knowledge graph identification improves on the baseline, the NELL promotion strategy, and the previous state-of-the-art results using MLNs, with a modest improvement in AUC and a substantial improvement in the F1 measure.

Analyzing Variations of Knowledge Graph Identification

To better understand the contributions of various components of our knowledge graph identification model, we explore variants that omit one aspect of the model: capturing different extraction sources, using entity coreference information, and the ontological constraints between facts. We compare the results for each of these treatments in table 3. PSL-NoSrcs removes predicates CandLbl$_T$ and CandRel$_T$ for different candidate sources, replacing them with a single predicate using the average confidence value across sources. PSL-NoER removes rules used to reason about coreferent entities, easing the constraint that coreferent entities share the same labels and relations. PSL-NoOnto removes rules for using ontological relationships to constrain the knowledge graph, removing many of the dependencies necessary for consistency. Removing source information and entity resolution each reduces the performance of our model slightly, but the large drop in AUC and F1 measure when ontological information is removed suggests that the ontology is the principal contributor to the success of knowledge graph identification.

One drawback of our comparisons to previous work is the restriction of the model to a small set of inference targets. The construction of this set obscures some of the challenges presented in real-world data, such as conflicting evidence. To assess the performance of our method in a setting where inference targets do not restrict potentially contradictory inferences, we also ran knowledge graph identification using the same learned weights but with no predefined set of targets, allowing lazy inference to produce a complete knowledge graph. The resulting inference produces a total of 4.9 million facts, which subsume the test set. We report results of this process in table 4 as PSL-KGI-Complete, using the same evaluation set as previous experiments. Allowing the model to optimize on the full knowledge graph instead of just the test set reduced the performance on the test set, suggesting that the noise introduced by conflicting evidence does have an impact on results. Despite the difficulties of inference in this more complex setting, running inference on the full knowledge graph improves AUC and F1 relative to the heuristics NELL uses to produce its knowledge base.

Method	AUC	F1
NELL	0.765	0.673
PSL-KGI	0.904	0.853
PSL-KGI-Complete	**0.892**	**0.848**

Table 4. Producing a Complete Knowledge Graph Reduces Performance on the Test Set.

This suggests that the true complexity of the problem is masked when generating a limited set of inferences.

Scalability

One advantage of using PSL for knowledge graph identification is the ability to frame complex joint reasoning as a convex optimization. Knowledge graph identification implemented in PSL can handle problems from real-world data sets like NELL, which include millions of candidate facts. Inference when an explicit query set of 70,000 facts is given (PSL-KGI) requires a mere 10 seconds. The MLN method we compare against takes a few minutes to an hour to run for the same setting. When inferring a complete knowledge graph without known query targets, as in the last NELL experiment, inference with MLNs is infeasible. In contrast, knowledge graph identification on the NELL data set can produce the complete knowledge graph containing 4.9 million facts in only 130 minutes. The ability to produce complete knowledge graphs in these realistic settings is an important feature of our implementation of knowledge graph identification.

Conclusion

Successfully combining statistical features and semantic relationships is a common theme in artificial intelligence research. In this article, we describe a statistical relational learning approach for combining these two disparate sources of knowledge. Specifically, we show how the noisy candidate facts and statistical features produced by an information-extraction system can be combined with semantic constraints derived from an ontology to produce a knowledge base. Using the knowledge graph representation for the knowledge base, we describe the problem of knowledge graph identification: jointly inferring the most likely knowledge graph. Our model for knowledge graph identification defines a probability distribution over possible knowledge graphs using a series of logical formulas. Scalability is a key concern for inference in joint models; however, using probabilistic soft logic allows us to solve the MPE inference problem through an efficient convex optimization. In our results on data from the NELL project, we demonstrate that knowledge graph identification is capable of producing superior knowledge graphs while scaling to problems that are intractable for competing models. In future work, we plan to extend knowledge graph identification to address constantly growing and changing web data.

Acknowledgments

We would like to thank Shangpu Jiang and Daniel Lowd for sharing their data and offering enthusiastic assistance. This work was partially supported by NSF CAREER grant 0746930, NSF grants IIS1218488 and CCF0937094, and Google Research grants. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Note

1. github.com/linqs/KnowledgeGraphIdentification.

References

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 *Advances in Neural Information Processing Systems,* volume 25, 2654–2662. Red Hook, NY: Curran Associates, Inc.

Broecheler, M.; Mihalkova, L.; and Getoor, L. 2010. Probabilistic Similarity Logic. In *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence,* 73–82. Corvallis, OR: AUAI Press.

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 *Proceedings of the 24th AAAI Conference on Artificial Intelligence*. Menlo Park, CA: AAAI Press.

Etzioni, O.; Banko, M.; Soderland, S.; and Weld, D. S. 2008. Open Information Extraction from the Web. *Communications of the ACM* 51 (12). dx.doi.org/10.1145/1409360.1409378

Getoor, L., and Taskar, B. 2007. *Introduction to Statistical Relational Learning*. Cambridge, MA: The MIT Press.

Hitzler, P., and van Harmelen, F. 2010. A Reasonable Semantic Web. *Semantic Web* 1(1): 39–44.

Jiang, S.; Lowd, D.; and Dou, D. 2012. Learning to Refine an Automatically Extracted Knowledge Base Using Markov Logic. In *Proceedings of the 12th International Conference on Data Mining*. Los Alamitos, CA: IEEE Computer Society.

Namata, G. M.; Kok, S.; and Getoor, L. 2011. Collective Graph Identification. In *Proceedings of 17th ACM SIGKDD Conference on Knowledge Discovery and Data Mining,* 87–95. New York: Association for Computing Machinery.

Niu, F.; Zhang, C.; Ré, C.; and Shavlik, J. 2012. DeepDive: Web-Scale Knowledge-Base Construction Using Statistical Learning and Inference. Paper presented at the Second International Workshop on Searching and Integrating New Web Data Sources. Colocated with VLDB 2012, Istanbul, Turkey, Aug. 31.

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 *Proceedings of the 21st AAAI Conference on Artificial Intelligence,* 1400–1405. Menlo Park, CA: AAAI Press.

Pujara, J.; Miao, H.; Getoor, L.; and Cohen, W. 2013. Knowledge Graph Identification. In *Semantic Web — ISWC 2012,* Lecture Notes in Computer Science volume 7650. 542–557. Berlin: Springer.

Suchanek, F. M.; Kasneci, G.; and Weikum, G. 2007. YAGO: A Core of Semantic Knowledge. In *Proceedings of the 16th International World Wide Web Conference,* 697–706. New York: Association for Computing Machinery. dx.doi.org/10.1145/1242572.1242667

Weikum, G., and Theobald, M. 2010. From Information to Knowledge: Harvesting Entities and Relationships from Web Sources. In *Proceedings of the 29th Symposium on Principles of Database Systems,* 65–76. New York: Association for Computing Machinery.

Wimalasuriya, D. C., and Dou, D. 2010. Ontology-Based Information Extraction: An Introduction and a Survey of Current Approaches. *Journal of Information Science* 36 (3): 306–323. dx.doi.org/10.1177/0165551509360123

Jay Pujara (www.cs.umd.edu/jay) is a Ph.D. student in the Computer Science Department at the University of Maryland, College Park. He received M.S. and B.S. degrees from Carnegie Mellon University. His research interests include machine learning, scalable statistical relational learning, and knowledge base construction, and he is the recipient of two best paper awards.

Hui Miao (www.cs.umd.edu/hui) is a Ph.D. student in the Computer Science Department at University of Maryland, College Park. He works on large-scale machine-learning algorithms and user-friendly database systems.

Lise Getoor is a professor in the Computer Science Department at the University of California, Santa Cruz. Her primary research interests are in machine learning and reasoning with uncertainty, applied to graphs and semistructured data. She has eight best paper and best-student paper awards and an NSF Career Award, was program committee cochair for the 2011 International Machine Learning Conference (ICML), and is an Association for the Advancement of Artificial Intelligence (AAAI) Fellow. She received her Ph.D. from Stanford University, her M.S. from the University of California, Berkeley, and her B.S. from the University of California, Santa Barbara.

William W. Cohen is a professor in the Machine Learning Department and Language Technologies Institute at Carnegie Mellon University. Cohen's research interests include information integration and machine learning, particularly information extraction, text categorization, and learning from large data sets. He holds seven patents related to learning, discovery, information retrieval, and data integration and is the author of more than 200 publications.