Knowledge Extraction and Joint Inference Using Tractable Markov Logic

Chloé Kiddon and **Pedro Domingos**
Department of Computer Science & Engineering
University of Washington
Seattle, WA 98105
{chloe,pedrod}@cs.washington.edu

Abstract

The development of knowledge base creation systems has mainly focused on information extraction without considering how to effectively reason over their databases of facts. One reason for this is that the inference required to learn a probabilistic knowledge base from text at any realistic scale is intractable. In this paper, we propose formulating the joint problem of fact extraction and probabilistic model learning in terms of Tractable Markov Logic (TML), a subset of Markov logic in which inference is low-order polynomial in the size of the knowledge base. Using TML, we can tractably extract new information from text while simultaneously learning a probabilistic knowledge base. We will also describe a testbed for our proposal: creating a biomedical knowledge base and making it available for querying on the Web.

1 Introduction

While structured sources of information exist, so much of human knowledge is found only in unstructured text that it is crucial we learn how to mine these unstructured sources efficiently and accurately. However, knowledge extraction is only half the battle. We develop knowledge bases not to be standalone structures but instead to be tools for applications such as decision making, question answering, and literature-based discovery. Therefore, a knowledge base should not be a static repository of facts; it should be a probabilistic model of knowledge extracted from text over which we can infer new facts not explicitly stated in the text.

Most current knowledge extraction systems extract a database of facts, not a true knowledge base. ReVerb (Etzioni et al., 2011) and TextRunner (Banko et al., 2007) are Web-scale knowledge extraction systems, but provide no clear method for reasoning over the extracted knowledge. Unsupervised Semantic Parsing (USP) and its successor Ontological USP (OntoUSP), learn more detailed ontological structure over information extracted from text, but they too do not build a coherent probabilistic knowledge base that can be reasoned with (Poon and Domingos 2009, Poon and Domingos 2010).

Some knowledge extraction systems have integrated rule learning. NELL learns rules to help extract more information, but the resulting knowledge base is still just a collection of facts (Carlson et al., 2010). The SHERLOCK system learns first-order Horn clauses from open-domain Web text, but the inferences allowed are not very deep and, like ReVerb and TextRunner, the database of facts is not structured into any useful ontology (Schoenmackers et al. 2008, Schoenmackers et al. 2010).

In this paper, we propose an unsupervised online approach to knowledge base construction that jointly extracts information from text and learns a probabilistic model of that information. For each input sentence, our approach will jointly learn the best syntactic and semantic parse for the sentence while using abductive reasoning to infer the changes to our knowledge base that best explain the information in the sentence. To keep this joint inference procedure tractable we will formulate our entire process in terms of Tractable Markov Logic. *Tractable Markov Logic (TML)* is a subset of Markov logic in

Proc. of the Joint Workshop on Automatic Knowledge Base Construction & Web-scale Knowledge Extraction (AKBC-WEKEX), pages 79–83,
NAACL-HLT, Montréal, Canada, June 7-8, 2012.

	Name	TML Syntax	Comments	Example
Rules	Subclass	$\texttt{Is}(\texttt{C}_1, \texttt{C}_2)\texttt{:w}$		$\texttt{Is}(\texttt{Lion}, \texttt{Mammal})$
	Subpart	$\texttt{Has}(\texttt{C}_1, \texttt{C}_2, \texttt{P}, \texttt{n})$	P, n optional	$\texttt{Has}(\texttt{EatingEvent}, \texttt{Animal}, \texttt{Eater})$
	Relation	$\texttt{R}(\texttt{C}, \texttt{P}_1, \ldots, \texttt{P}_\texttt{n})\texttt{:w}$	$\neg\texttt{R}(\ldots)$ allowed	$\texttt{Eats}(\texttt{EatingEvent}, \texttt{Eater}, \texttt{Eaten})$
Facts	Subclass	$\texttt{Is}(\texttt{X}, \texttt{C})$	$\neg\texttt{Is}(\texttt{X}, \texttt{C})$ allowed	$\texttt{Is}(\texttt{Simba}, \texttt{Lion})$
	Subpart	$\texttt{Has}(\texttt{X}_1, \texttt{X}_2, \texttt{P})$		$\texttt{Has}(\texttt{TheLionKing}, \texttt{Simba}, \texttt{Protagonist})$
	Relation	$\texttt{R}(\texttt{X}, \texttt{P}_1, \ldots, \texttt{P}_\texttt{n})$	$\neg\texttt{R}(\ldots)$ allowed	$\texttt{Defeats}(\texttt{TheLionKing}, \texttt{Simba}, \texttt{Scar})$

Table 1: The TML language

which exact inference is low-order polynomial in the size of the knowledge base (Domingos and Webb, 2012). TML is a surprisingly powerful language that can easily represent both semantic relations and facts and syntactic relations.

2 Tractable Markov Logic

Tractable Markov Logic (TML) (Domingos and Webb, 2012), is a tractable, yet quite powerful, subset of Markov logic, a first-order probabilistic language. A *Markov logic network (MLN)* is a set of weighted first-order logic clauses (Domingos and Lowd, 2009). Given a set of constants, an MLN defines a Markov network with one node per ground atom and one feature per ground clause. The weight of a feature is the weight of the first-order clause that originated it. The probability of a state $\mathbf{x}$ is given by $P(\mathbf{x}) = \frac{1}{Z}\exp\left(\sum_i w_i n_i(\mathbf{x})\right)$, where w_i is the weight of the i^{th} clause, and n_i is the number of satisfied groundings of that clause. $Z = \sum_{\mathbf{x}} \exp(\sum_i w_i n_i(\mathbf{x}))$ is the partition function. A *TML knowledge base (KB)* is a set of rules with three different forms, summarized in Table 1. A TML rule $\texttt{F} : w$ states that formula F has weight w. The conversion from rules in TML syntax to clauses in MLN syntax is straightforward. For details, see Webb and Domingos 2012.

Subclass rules define the hierarchy of classes in the TML KB. *Subpart rules* define decompositions of the part classes in the TML KB into their subpart classes. *Relation rules* define arbitrary relations between the subparts of a given class. There are three types of corresponding facts in TML that provide information about objects instead of classes: the classes of objects, the objects that are subparts of other objects, and relations between objects. Naturally, the facts in TML must be consistent with the structure set by the TML rules for the KB to be valid. For example, a fact can not define a subpart relation between two objects if that subpart relation does not exist as a rule between the classes of those objects.

There are a number of constraints on a set of TML rules for it to be a valid TML KB. The class hierarchy must be a forest, and subclasses of the same class are mutually exclusive. Also, the polarity of ground literals must be consistent among the descendants of an object's subparts under the same class. However, given these restrictions on the form of the TML KB, Theorem 1 of Domingos and Webb 2012 states that the partition function of a TML KB can be computed in time and space polynomial in the size of the knowledge base. The intuition behind this theorem is that traversing structure of the class hierarchy and part decomposition of the TML KB is isomorphic to the computation of the partition function of the corresponding MLN. Since the probability of a query can be computed as a ratio of partition functions, computing it is also tractable.

At first glance, it may seem that TML is a very restrictive language. However, TML is surprisingly flexible; it can compactly represent arbitrary junction trees and many high-treewidth models. The cost of using TML is that it cannot tractably represent all arbitrary networks, especially those with many dependencies between related objects (Domingos and Webb, 2012). However, when a network contains hierarchical structure, with bounds on the number of links between objects in different classes, the TML KB remains tractable. As shown in the success of OntoUSP, many statements in natural language can be semantically parsed into a hierarchical part/class structure. Syntax also has this kind of structure; smaller syntactic components form the subparts for larger components. We will now briefly describe how TML is a very natural fit for both the syntactic and semantic realms.

2.1 TML for syntactic parsing

Non-recursive probabilistic context-free grammars (PCFGs) (Chi, 1999) can be compactly encoded in TML. Non-terminals have class-subclass relationships to their set of productions. Each production is split into subparts based on the symbols appearing on its right-hand side. It is straightforward to show how to transform one of these grammars into a TML KB. (For a proof sketch see Domingos and Webb 2012.) Natural language is recursive, but fixing the number of recursive levels will allow for a grammar flexible enough for virtually all real sentences. Once we have the PCFG encoded in TML, we can find the most likely parse of a sentence using the standard TML inference algorithm.

2.2 TML for semantic parsing

TML closely mirrors the ontological structure of objects in the world. Objects are defined by class structure (e.g., monkeys are mammals), part decompositions (e.g., monkeys have a tail, legs, etc.), and relations (e.g., a monkey's tail is between its legs).

Text also frequently contains relations occurring between objects. These relations and constructs in natural language contain rich ontological structure; we hypothesize that this structure allows TML to compactly represent semantic information about relations and events. For example, to describe the food chain, we define a class for the eating relation with two subparts: the eater of the animal class and the eaten of the living thing class. This eating relation class would have subclasses to define carnivorous and vegetarian eating events and so on, refining the subpart classes as needed. Since animals tend to only eat things of one other class, the number of eating relation classes will be low, and the TML can tractably represent these relations. This approach can be easily extended to a hierarchy of narrative classes, which each contain up to a fixed number of events as subparts.

TML can also be used to deal with other types of phenomena in natural language. (Space precludes us from going into detail for many here.) For example, adding place markers to a TML KB is straightforward. A class can have a location subpart whose class is selected from a hierarchy of places.

3 TML for knowledge base construction

To create a knowledge base from unstructured text, we propose a joint inference procedure that takes as input a corpus of unstructured text and creates a TML knowledge base from information extracted from the text. For each sentence, this inference procedure will jointly find the maximum a posteriori (MAP) syntactic and semantic parse and abduce the best changes to the TML KB that explain the knowledge contained in the sentence. Unlike previous pipeline-based approaches to knowledge base construction where a sentence is parsed, facts are extracted, and a knowledge base is then induced, we propose to do the whole process jointly and online. As we infer the best parses of sentences, we are simultaneously learning a probabilistic model of the world, in terms of both structure and parameters.

We plan to develop our approach in stages. At first, we will take advantage of existing syntactic and semantic parsers (e.g., an existing PCFG parser + USP) to parse the text before converting to TML. We may also bootstrap our KB from existing ontologies. However, we will steadily integrate more of the parsing into the joint framework by replacing USP with a semantic parser that parses text straight into TML, and eventually replacing the syntactic parser with one formulated entirely in TML.

3.1 Inference

The probability of a joint syntactic parse T and semantic parse L for a sentence S using a TML KB K is $Pr(T, L|S) \propto \exp(\sum_i w_i n_i(T, L, S))$, where the sum is indexed over the clauses in the MLN created from converting K into Markov logic. Exact MAP inference is possible in MLNs formed from TML KBs. Therefore, finding the joint MAP syntactic and semantic parse for a sentence with a parser formulated as a TML KB is tractable. The tractability of inference is vital since the MAP parse of a sentence given a current state of the TML KB will need to be found frequently during learning.

Inference in a TML KB is low-order polynomial in the size of the KB. However, if the size becomes exponential, inference will no longer be tractable. In this case, we can utilize variational inference to approximate the intractable KB with the closest tractable one (Lowd and Domingos, 2010). How-

ever, in general, even if the full KB is intractable, the subset required to answer a particular query may be tractable, or at least easier to approximate.

3.2 Learning

As we parse sentences, we simultaneously learn the best TML KB that explains the information in the sentences. Given the MAP parse, the weights for the KB can be re-estimated in closed form by storing counts from previously-parsed knowledge and by using m-estimation for smoothing among the classes. However, we also need to search over possible changes to the part and class structure of the KB to find the state of the KB that best explains the parse of the sentence. Developing this structure search will be a key focus of our research.

We plan to take advantage of the fact that sentences tend to either state general rules (e.g., "Penguins are flightless birds") or facts about particular objects (e.g., "Tux can't fly"). When parsing a sentence that states a general rule, the structure learning focuses on how best to alter the class hierarchy or part decomposition to include the new rule and maintain a coherent structure. For example, parsing the sentence about penguins might involve adding penguins as a class of birds and updating the weight of the `CanFly(c)` relation for penguins, which in turn changes the weight of that relation for birds. For sentences that state properties or relations on objects, learning will involve identifying (or creating) the best classes for the objects and updating the weight of the property or relation involved. When learning, we will have to ensure that no constraints of the TML KB are violated (e.g., the class hierarchy must remain a forest).

3.3 Querying the database

Inferring the answer of a yes/no query is simply a matter of parsing a query, adding its semantic parse to the KB, and recomputing the partition function (which is tractable in TML). The probability of the query is the value of the new partition function divided by the old. For more substantive queries (e.g., "What does IL-13 enhance?"), the naïve approach would look at each possible answer in turn. However, we can greatly speed up this process using coarse-to-fine inference utilizing the class structure of the TML KB (Kiddon and Domingos, 2011).

4 Proposed testbed

As an initial testbed, we plan to use our approach to build a knowledge base from the text of PubMed[1] and PubMed Central[2], companion repositories of 21 million abstracts and 2.4 million full texts of biomedical articles respectively. PubMed is a good basis for an initial investigation of our methods for a number of reasons. A biomedical knowledge base is of real use and importance for biomedical researchers. PubMed is a good size: large and rich, but not Web-scale, which would require parallelization techniques beyond our proposal's scope. Also, since the repositories contain both abstracts and full-text articles, we can incrementally scale up our approach from abstracts to full text articles, until eventually extracting from both repositories. The biomedical domain is also a good since shallow understanding is attainable without requiring much domain knowledge. However, if needed, we can seed the knowledge base with information extracted from biology textbooks, biology ontologies, etc.

There will be many questions our KB cannot answer, but even if we are far from solving the knowledge extraction problem, we can do much better than the existing keyword-based retrieval offered by the repositories. We also plan to go further with our proposal and make our knowledge base available for querying on the Web to allow for peer evaluation.

5 Conclusion

We propose an approach to automatic knowledge base construction based on using tractable joint inference formulated in terms of Tractable Markov Logic. Using TML is a promising avenue for extracting and reasoning over knowledge from text, since it can easily represent many kinds of syntactic and semantic information. We do not expect TML to be good at everything, and a key part of our research agenda is discovering which language extraction and understanding tasks it is good at and which may need additional methods. We plan to use biomedical texts as a testbed so we may see how a knowledge base created using our approach performs in a large, real-world domain.

[1] http://www.ncbi.nlm.nih.gov/pubmed/

[2] http://www.ncbi.nlm.nih.gov/pmc/

Acknowledgments

This research was partly funded by ARO grant W911NF-08-1-0242, AFRL contract FA8750-09-C-0181, NSF grant IIS-0803481, ONR grant N00014-08-1-0670, and the National Science Foundation Graduate Research Fellowship under Grant No. DGE-0718124. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ARO, DARPA, AFRL, NSF, ONR, or the United States Government.

References

M. Banko, M. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. 2007. Open information extraction from the web. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence*, pages 2670–2676.

A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E.R. Hruschka Jr., and T.M. Mitchell. 2010. Toward an architecture for never-ending language learning. In *Proceedings of the Twenty-Fourth National Conference on Artificial Intelligence*, pages 1306–1313.

Z. Chi. 1999. Statistical properties of probabilistic context-free grammars. *Computational Linguistics*, 25:131–160.

P. Domingos and D. Lowd. 2009. *Markov Logic: An Interface Layer for Artificial Intelligence*. Morgan Kaufmann.

P. Domingos and W. A. Webb. 2012. A tractable first-order probabilistic logic. In *Proceedings of the Twenty-Sixth National Conference on Artificial Intelligence*.

O. Etzioni, A. Fader, J. Christensen, S. Soderland, and Mausam. 2011. Open information extraction: The second generation. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, pages 3–10.

C. Kiddon and P. Domingos. 2011. Coarse-to-fine inference and learning for first-order probabilistic models. In *Proceedings of the Twenty-Fifth National Conference on Artificial Intelligence*, pages 1049–1056.

D. Lowd and P. Domingos. 2010. Approximate inference by compilation to arithmetic circuits. In *Advances in Neural Information Processing Systems*, pages 1477–1485.

H. Poon and P. Domingos. 2009. Unsupervised semantic parsing. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1–10.

H. Poon and P. Domingos. 2010. Unsupervised ontology induction from text. In *Proceedings of the Association for Computational Linguistics*, pages 296–305.

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

S. Schoenmackers, O. Etzioni, D. Weld, and J. Davis. 2010. Learning first-order horn clauses from web text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1088–1098.