Machine Reading: A "Killer App" for Statistical Relational AI

Hoifung Poon **Pedro Domingos**
Department of Computer Science and Engineering
University of Washington
Seattle, WA 98195-2350, U.S.A.
{*hoifung, pedrod*}@*cs.washington.edu*

Abstract

Machine reading aims to automatically extract knowledge from text. It is a long-standing goal of AI and holds the promise of revolutionizing Web search and other fields. In this paper, we analyze the core challenges of machine reading and show that statistical relational AI is particularly well suited to address these challenges. We then propose a unifying approach to machine reading in which statistical relational AI plays a central role. Finally, we demonstrate the promise of this approach by presenting OntoUSP, an end-to-end machine reading system that builds on recent advances in statistical relational AI and greatly outperforms state-of-the-art systems in a task of extracting knowledge from biomedical abstracts and answering questions.

Introduction

Machine reading aims to extract knowledge from unstructured text with little human effort. It has been a major goal of AI since its early days. The advent of the Web makes available billions of documents and virtually unlimited amount of knowledge to extract, further increasing the importance and urgency of machine reading. The success of machine reading will not only help breach the knowledge acquisition bottleneck in AI, but also revolutionize Web search, scientific and applied research (e.g., biomedical research and drug design (Poon and Vanderwende 2010)), and other fields.

In the past, there has been a lot of progress in automating many subtasks of machine reading by machine learning approaches (e.g., components in the traditional NLP pipeline such as tagging and parsing). However, end-to-end solutions are still rare, and existing systems typically require substantial amount of human effort in manual engineering and/or labeling examples. As a result, they often target restricted domains and only extract limited types of knowledge (e.g., a pre-specified relation). Moreover, many machine reading systems train their knowledge extractors once and do not leverage further learning opportunities such as additional text and interaction with end users.

Ideally, a machine reading system should strive to satisfy the following desiderata:

End-to-end: the system should input raw text, extract knowledge, and be able to answer questions and support other end tasks;

High quality: the system should extract knowledge with high accuracy;

Large-scale: the system should acquire knowledge at Web-scale and be open to arbitrary domains, genres, and languages;

Maximally autonomous: the system should incur minimal human effort;

Continuous learning from experience: the system should constantly integrate new information sources (e.g., new text documents) and learn from user questions and feedback (e.g., via performing end tasks) to continuously improve its performance.

These desiderata raise many challenging research questions. In this paper, we argue that the key to resolving such challenges hinges on joint inference and uncertainty handling, the combination of which is the hallmark of statistical relational AI. Therefore, machine reading is a natural "killer app" for statistical relational AI. In addition, we propose a unifying approach to machine reading that capitalizes on recent advances in statistical relational AI.

To demonstrate the promise of this approach, we present OntoUSP (Poon and Domingos 2010), an end-to-end machine reading system that features large-scale joint inference. In a task of extracting knowledge from biomedical abstracts and answering questions, it greatly outperforms state-of-the-art approaches in both precision and recall. For example, compared to TextRunner (Banko et al. 2007), the state-of-the-art open information-extraction system, OntoUSP not only improved precision by over 38 points, but also extracted more than five times of correct answers.

We begin by discussing the core challenges in machine reading. We then proposing our unifying approach based on statistical relational AI. Finally, we present the OntoUSP system and the experimental result.

Machine Reading: Core Challenges

A salient challenge to machine reading is the prevailing uncertainty in text understanding. Linguistic analyses are highly ambiguous at all levels from morphology to pragmatics. Moreover, contradictions and errors abound in natural

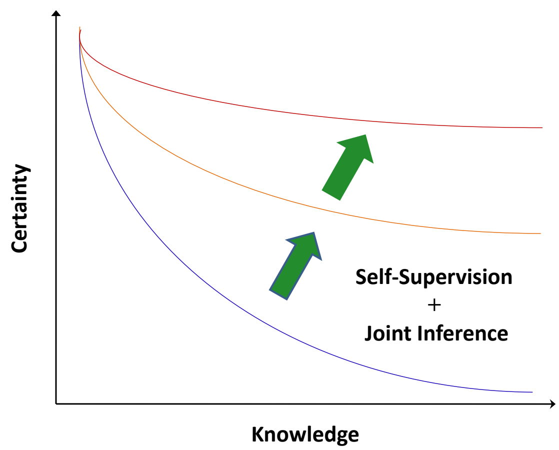

Figure 1: A unifying vision for machine reading: bootstrap from the head regime of the power-law distribution of textual knowledge, and conquer the long tail in a self-supervised learning process that raises certainty on sparse extractions by propagating information via joint inference from frequent extractions.

text. Consequently, a machine reading system must be well versed in handling uncertainty.

Another salient challenge is *the long tail of textual knowledge*. The heterogeneous Web contains texts that vary substantially in subject matters (e.g., finance vs. biology) and writing styles (e.g., blog posts vs. scientific papers). In addition, natural languages are famous for their myraid variations in expressing the same meaning. A fact may be stated in a straightforward way such as "kale contains calcium". More often though, it may be stated in a syntactically and/or lexically different way than as phrased in an end task (e.g., "calcium is found in kale"). Finally, many facts are not even stated explicitly, and must be inferred from other facts (e.g., "kale prevents osteoporosis" may not be stated explicitly but can be inferred by combining facts such as "kale contains calcium" and "calcium helps prevent osteoporosis").

As a result, a machine reading system must not rely on explicit supervision such as manual rules and labeled examples, which will incur prohibitive cost in the Web scale. Instead, it must be able to learn from indirect supervision. A key source of indirect supervision is redundancy (Downey, Etzioni, and Soderland 2010). While a rare extraction may arise by chance of error, it is much less likely so for the ones with many repetitions. Such highly-redundant knowledge can be extracted easily and with high confidence, and can be leveraged for bootstrapping.

For knowledge that resides in the long tail, explicit forms of redundancy (e.g., identical expressions) are rare, but this can be circumvented by joint inference. For example, expressions that are composed with or by similar expressions probably have the same meaning. In effect, joint inference abandons the i.i.d. assumption and leverages the interdependencies to propagate information among objects and relations (Getoor and Taskar 2007; Bakir et al. 2007).

In general, joint inference can take various forms, ranging from simple voting to sophisticated probabilistic reasoning over a joint model. Simple ones tend to scale better, but their capability in propagating information is limited. Sophisticated methods can uncover implicit redundancy and propagate much more information with higher quality, yet the challenge is how to make them scale as well as simple ones. A self-supervised learning process stipulates what form of joint inference to use and how. Effectively, it increases certainty on sparse extractions by propagating information from more frequent ones. Figure 1 illustrates this vision for machine reading.

Recently, there has been increasing interest in incorporating active human involvement into machine learning processes (Basu and Kapoor 2009). Machine reading can naturally benefit from this direction. The output of a machine reading system (i.e., the extracted knowledge) can be applied to end tasks such as question answering, whereas the interactions with human in performing such end tasks provide valuable indirect supervision (e.g., whether users find the answer useful or not). With joint inference, such feedback can be easily incorporated into the reading system to improve performance.

A Unifying Approach to Machine Reading

In the previous section, we observe that joint inference is essential for machine reading. Recent research in statistical relational AI offers a plethora of approaches for joint inference and learning (Getoor and Taskar 2007; Bakir et al. 2007). We propose to use Markov logic (Domingos and Lowd 2009) for machine reading since it is the leading unifying framework for statistical relational AI, but other approaches can be used as well. Markov logic is a probabilistic extension of first-order logic and can compactly specify probability distributions over complex relational domains. It has been successfully applied to unsupervised learning for various NLP tasks such as coreference resolution (Poon and Domingos 2008) and semantic parsing (Poon and Domingos 2009). A *Markov logic network (MLN)* is a set of weighted first-order clauses. Together with a set of constants, it 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 x in such a network is given by the log-linear model $P(x) = \frac{1}{Z}\exp\left(\sum_i w_i n_i(x)\right)$, where Z is a normalization constant, w_i is the weight of the ith formula, and n_i is the number of satisfied groundings.

While joint inference can make machine reading more effective, scaling up joint inference remains a prominent challenge. We propose to use coarse-to-fine inference (Felzenszwalb and McAllester 2007; Petrov 2009; Kiddon and Domingos 2010) as a unifying framework to scale joint inference to the Web. Essentially, coarse-to-fine inference leverages the sparsity imposed by hierarchical structures that are ubiquitous in human knowledge (e.g., taxonomies/ontologies). At coarse levels, ambiguities are rare (there are few objects and relations), and inference can be conducted efficiently. The result is then used to prune unpromising refinements at the next level. This process continues down the hierarchy until decision can be made. In this way, inference can potentially speed up exponentially, analogous to binary tree search.

The success of coarse-to-fine inference hinges on the availability of appropriate hierarchical structures. An ontology specifies entities and their relations in a problem domain, among which are the ISA and ISPART hierarchies that can be used to support coarse-to-fine inference. In general, constructing the ontology (ontology induction) and mapping textual expressions to ontological nodes (ontology population) remain difficult open problems (Staab and Studer 2004). Traditional approaches are manual, which makes them very costly and limits them to well-circumscribed domains. More recently, machine learning approaches have been developed (e.g., Snow et al. (2006), Cimiano (2006), Suchanek et al. (2008,2009), Wu & Weld (2008)), but they are still limited in several aspects. First, many approaches induce and populate a deterministic ontology, which does not capture the inherent uncertainty among the entities and relations. Second, most approaches either bootstrap from heuristic patterns (e.g., Hearst patterns (Hearst 1992)) or build on existing structured or semi-structured knowledge bases (e.g., WordNet (Fellbaum 1998) or Wikipedia), and thus have limited coverage. Third, most approaches focus on inducing ontology over words rather than arbitrarily large meaning units (e.g., idioms, phrasal verbs, etc.). Finally and most important of all, existing approaches typically separate ontology induction from population and knowledge extraction, and pursue each task independently. This is highly suboptimal. The resulted ontology is disconnected from text and requires additional effort to map between the two. Moreover, this fails to leverage the intimate connections between the three tasks for mutual disambiguiation and propagating information.

Therefore, we propose to induce a probabilistic ontology from text, and to do so jointly with ontology population and knowledge extraction. A probability model governs the joint process which simultaneously does the following: parse text and identifies meaning units (extraction), assign them to existing ontological nodes (population) and/or create new nodes (induction). Such a joint approach can unlock much more implicit information in text to help ontology induction and population, and can potentially work well even in domains not equipped with resources like WordNet and Wikipedia (e.g., biomedical papers). Furthermore, we propose to incorporate hierarchical smoothing into the induction process, which enables more accurate parameter estimation and better generalization. To support coarse-to-fine inference, we propose to inject some learning bias (e.g., in the form of priors) to favor inducing an ontology that facilitates efficient coarse-to-fine inference.

To maximize the benefit from interactive machine learning, we propose a novel form of continuous learning that leverages the interaction with end users to constantly improve the system performance. This is straightforward to do via self-supervision and joint inference. Essentially, when the system output is applied to an end task (e.g., answering questions), the user feedback is incorporated into the system as a bootstrap resource. Feedback can take the form of explicit supervision (e.g., via community content creation or active learning) or indirect signals (e.g., click data and query logs). We thus bootstrap an online community from an initial machine reading system that provides imperfect but valuable end services, and continuously improve the quality of system output to attract more users with higher degree of participation, thereby creating a positive feedback loop and raising the machine reading performance to a high level that is difficult to attain otherwise.

In sum, we propose the following unifying approach to machine reading:

1. Bootstrap from easiest extractable knowledge and conquer the long tail via a self-supervised learning process.
2. Apply Markov logic as the unifying framework for knowledge representation and joint inference.
3. Govern self-supervised learning by a probabilistic model that incorporates a small amount of heuristic knowledge with large-scale joint inference to maximize the amount and quality of information to propagate.
4. Scale up joint inference with coarse-to-fine inference.
5. Automatically induce probabilistic ontologies from text for better generalization and tractable coarse-to-fine inference. The ontology induction and population are incorporated into the probabilistic model for self-supervision.
6. Accomplish continuous learning by combining *bootstrapping* and *crowdsourced content creation* to synergistically improve the reading system from user interaction.

OntoUSP

In this section, we demonstrate the promise of our unifying approach by presenting OntoUSP (Poon and Domingos 2010), a recent system that capitalizes on Markov logic and joint inference and achieves state-of-the-art results on a machine reading task. OntoUSP (Poon and Domingos 2010) builds on the USP unsupervised semantic parser (Poon and Domingos 2009) and synergistically conducts probabilistic ontology induction, population, knowledge extraction, and hierarchical smoothing in a single integrated process. In the remainder of the section, we first introduce the task of semantic parsing and the USP system. We then present OntoUSP and the experimental results. (Additional details can be found in Poon & Domingos (2010).)

Semantic Parsing

Semantic parsing aims to obtain a complete formal meaning representation for input sentences. It can be viewed as a structured prediction problem, where a semantic parse is formed by partitioning the input sentence (or a syntactic analysis such as a dependency tree) into meaning units and assigning each unit to an entity or relation (Figure 2). In effect, a semantic parser extracts knowledge from input text and converts them into logical forms (the semantic parses), which can be used in logical and probabilistic inference and support end tasks such as answering questions.

As discussed earlier, a major challenge facing semantic parsing is syntactic and lexical variations of the same meaning. For example, the fact that IL-4 protein induces CD11b can be expressed in a variety of ways, such as, “Interleukin-4 enhances the gene expression of CD11b”, “CD11b is upregulated by IL-4”, etc.

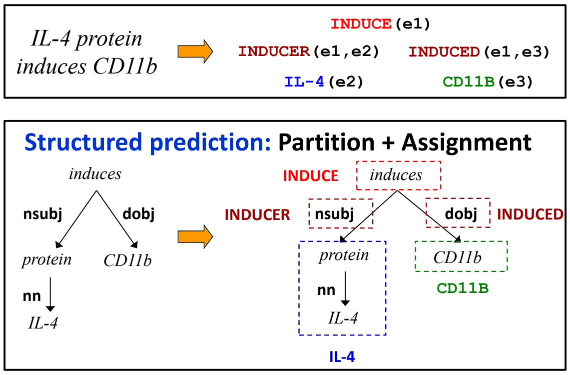

Figure 2: An example of semantic parsing. Top: semantic parsing converts an input sentence into logical forms (here in Davidsonian semantics). Bottom: a semantic parse consists of a partition of the dependency tree of the input sentence and the assignment of its parts.

Unsupervised Semantic Parsing

Past approaches to semantic parsing either manually construct a grammar or require example sentences with meaning annotation, and do not scale beyond restricted domains. Recently, we developed USP, the first unsupervised approach for semantic parsing (Poon and Domingos 2009).[1] USP inputs dependency trees of sentences and first transforms them into quasi-logical forms (QLFs). For each sentence, a semantic parse comprises of a partition of its QLF into lambda-form subexpressions, and an assignment of each subexpression to a *lambda-form cluster*. An *object cluster* corresponds to semantic concepts such as INDUCE, and contains a number of *property clusters* such as INDUCED (the patient argument of INDUCE). Each property cluster may in turn contains subclusters of property values (e.g., the patient argument-form subcluster may contain dependency labels like dobj and nsubjpass). Effectively, USP simultaneously discovers the lambda-form clusters and learns an IS-PART hierarchy among them. It does so by recursively combining subexpressions that are composed with or by similar subexpressions. The partition breaks a sentence into meaning units, and the clustering abstracts away syntactic and lexical variations for the same meaning.

This novel form of relational clustering is governed by a joint probability distribution $P(T, L)$ defined in Markov logic (Domingos and Lowd 2009), where T are input dependency trees, and L semantic parses. The predicates are:

$\mathtt{SubExpr}(\mathtt{s}, \mathtt{e})$: s is a subexpression of e;

$\mathtt{HasValue}(\mathtt{s}, \mathtt{v})$: s is of value v;

$\mathtt{IsPart}(\mathtt{c}, \mathtt{i}, \mathtt{p})$: p is a property cluster in object cluster c indexed by i.

The probability model of USP is captured by two formulas:

$$\mathtt{x} \in +\mathtt{p} \wedge \mathtt{HasValue}(\mathtt{x}, +\mathtt{v})$$
$$\mathtt{e} \in \mathtt{c} \wedge \mathtt{SubExpr}(\mathtt{x}, \mathtt{e}) \wedge \mathtt{x} \in \mathtt{p} \Rightarrow \exists \mathtt{i}.\mathtt{IsPart}(\mathtt{c}, \mathtt{i}, \mathtt{p}).$$

[1] In this paper, we use a slightly different formulation to facilitate the exposition of OntoUSP.

All free variables are implicitly universally quantified. The "+" notation signifies that the MLN contains an instance of the formula, with a separate weight, for each value combination of the variables with a plus sign. The first formula is the core of the model and represents the mixture of property values given the cluster. The second formula ensures that a property cluster must be a part in the corresponding object cluster; it is a hard constraint, as signified by the period at the end. To encourage clustering, USP imposes an exponential prior over the number of parameters.

To parse a new sentence, USP starts by partitioning the QLF into atomic forms, then hill-climbs on the probability using a search operator based on lambda reduction until it finds the MAP parse. In learning, USP starts with clusters of atomic lamba forms, maintains the optimal semantic parses according to current parameters, and hill-climbs on the log-likelihood of observed QLFs using two search operators:

$\mathtt{MERGE}(\mathtt{c}_1, \mathtt{c}_2)$ merges clusters $\mathtt{c}_1, \mathtt{c}_2$ into a larger cluster $\mathtt{c}$ by merging the core-form clusters and argument clusters of $\mathtt{c}_1, \mathtt{c}_2$, respectively (e.g., $\mathtt{c}_1 = \{\text{"}induce\text{"}\}$, $\mathtt{c}_2 = \{\text{"}enhance\text{"}\}$, and $\mathtt{c} = \{\text{"}induce\text{"}, \text{"}enhance\text{"}\}$.)

$\mathtt{COMPOSE}(\mathtt{c}_1, \mathtt{c}_2)$ creates a new lambda-form cluster c formed by composing the lambda forms in $\mathtt{c}_1, \mathtt{c}_2$ into larger ones (e.g., $\mathtt{c}_1 = \{\text{"}amino\text{"}\}$, $\mathtt{c}_2 = \{\text{"}acid\text{"}\}$, and $\mathtt{c} = \{\text{"}amino\ \ acid\text{"}\}$.)

Each time, USP executes the highest-scored operator and reparses affected sentences using the new parameters. The output contains the optimal clusters and parameters, as well as the MAP semantic parses of input sentences.

Unsupervised Ontology Induction From Text

A major limitation of USP is that it either merges two object clusters into one, or leaves them separate. This is suboptimal, because different object clusters may still possess substantial commonalities. Modeling these can help extract more general knowledge and answer many more questions. The best way to capture such commonalities is by forming an ISA hierarchy among the clusters. For example, INDUCE and INHIBIT are both subconcepts of REGULATE. Learning these ISA relations helps answer questions like "What regulates CD11b?" when the text states that "IL-4 induces CD11b" or "AP-1 suppresses CD11b".

For parameter learning, this is also undesirable. Without the hierarchical structure, each cluster estimates its parameters solely based on its own observations, which can be extremely sparse. The better solution is to leverage the hierarchical structure for smoothing (McCallum et al. 1998; Gelman and Hill 2006). For example, if we learn that "super-induce" is a verb and that in general verbs have active and passive voices, then even though "super-induce" only shows up once in the corpus as in "AP-1 is super-induced by IL-4", by smoothing we can still infer that this probably means the same as "IL-4 super-induces AP-1", which in turn helps answer questions like "What super-induces AP-1".

OntoUSP overcomes the limitations of USP by replacing the flat clustering process with a hierarchical clustering one, and learns an ISA hierarchy of lambda-form clusters in addition to the IS-PART one. The output of OntoUSP consists of an ontology, a semantic parser, and the MAP parses.

The OntoUSP MLN can be obtained by modifying the USP MLN with three simple changes. First, we introduce a new predicate $\mathtt{IsA}(\mathtt{c}_1, \mathtt{c}_2)$, which is true if cluster $\mathtt{c}_1$ is a subconcept of $\mathtt{c}_2$. For convenience, we stipulate that $\mathtt{IsA}$ is reflexive (i.e., $\mathtt{IsA}(\mathtt{c}, \mathtt{c})$ is true for any $\mathtt{c}$). Second, we add the following formulas to the MLN:

$$\mathtt{IsA}(\mathtt{c}_1, \mathtt{c}_2) \wedge \mathtt{IsA}(\mathtt{c}_2, \mathtt{c}_3) \Rightarrow \mathtt{IsA}(\mathtt{c}_1, \mathtt{c}_3).$$

$$\mathtt{IsA}(\mathtt{c}_1, \mathtt{c}_2) \wedge \mathtt{IsPart}(\mathtt{c}_1, \mathtt{i}, \mathtt{p}_1) \wedge \mathtt{IsPart}(\mathtt{c}_2, \mathtt{i}, \mathtt{p}_2) \Rightarrow \mathtt{IsA}(\mathtt{p}_1, \mathtt{p}_2).$$

The first formula simply enforces the transitivity of ISA relation. The second formula states that if the ISA relation holds for a pair of object clusters, it also holds between their corresponding property clusters. Both are hard constraints. Third, we introduce hierarchical smoothing into the model by replacing the USP mixture formula $\mathtt{x} \in +\mathtt{p} \wedge \mathtt{HasValue}(\mathtt{x}, +\mathtt{v})$ with a new formula

$$\mathtt{ISA}(\mathtt{p}_1, +\mathtt{p}_2) \ \wedge \ \mathtt{x} \in \mathtt{p}_1 \wedge \mathtt{HasValue}(\mathtt{x}, +\mathtt{v}).$$

Intuitively, for each $\mathtt{p}_2$, the weight corresponds to the delta in log-probability of $\mathtt{v}$ comparing to the prediction according to all ancestors of $\mathtt{p}_2$. The effect of this change is that now the value $\mathtt{v}$ of a subexpression $\mathtt{x}$ is not solely determined by its property cluster $\mathtt{p}_1$, but is also smoothed by statistics of all $\mathtt{p}_2$ that are super clusters of $\mathtt{p}_1$. Shrinkage takes place via interaction among these weights. In particular, if the weights for some property cluster $\mathtt{p}$ are all zero, it means that values in $\mathtt{p}$ are completely predicted by $\mathtt{p}$'s ancestors. In effect, $\mathtt{p}$ is backed off to its parent. To encourage shrinkage, we applied m-estimate when estimating the weights.

OntoUSP uses the same inference algorithm as USP, except that the MLN is different and so the probability may be different. For learning, it augments the USP algorithm with a few changes. Most important of all, besides from MERGE and COMPOSE, OntoUSP uses a third operator $\mathtt{ABSTRACT}(\mathtt{c}_1, \mathtt{c}_2)$, which does the following:

1. Create a new cluster $\mathtt{c}$;
2. Retain $\mathtt{c}_1, \mathtt{c}_2$; add ISA links from them to $\mathtt{c}$;
3. Create super-clusters for property clusters of c_1 and c_2 and place them in $\mathtt{c}$, so as to maximize the log-likelihood.

Intuitively, $\mathtt{c}$ corresponds to a more abstract concept that summarizes similar properties in $\mathtt{c}_\mathtt{i}$'s. Specifically, for property clusters $\mathtt{p}_1, \mathtt{p}_2$ in $\mathtt{c}_1, \mathtt{c}_2$ respectively, ABSTRACT may create a property cluster $\mathtt{p}$ with ISA links from $\mathtt{p}_\mathtt{i}$'s to $\mathtt{p}$.

Experiments We applied OntoUSP to extract knowledge from the GENIA dataset (Kim et al. 2003) and answer questions, and we evaluated it on the number of extracted answers and accuracy. GENIA contains 1999 PubMed abstracts. The question set contains 2000 questions that were created by sampling verbs and entities according to their frequencies in GENIA. Sample questions include "What regulates MIP-1alpha?", "What does anti-STAT 1 inhibit?".[2] We used these simple question types to focus the evaluation on knowledge extraction rather than engineering for specific question types or reasoning.

[2] `http://alchemy.cs.washington.edu/papers/-poon09`.

Table 1: Comparison of question answering results on the GENIA dataset.

	# Total	# Correct	Accuracy
KW	150	67	45%
KW-SYN	87	67	77%
TR-EXACT	29	23	79%
TR-SUB	152	81	53%
RS-EXACT	53	24	45%
RS-SUB	196	81	41%
DIRT	159	94	59%
USP	334	295	88%
OntoUSP	**480**	**435**	**91%**

OntoUSP is the first unsupervised approach that synergistically conducts ontology induction, population, and knowledge extraction. The systems closest in aim and capability are USP and TextRunner (Banko et al. 2007). We thus compared OntoUSP with them. Other systems that we compared with include: a keyword-matching baseline (KW), RESOLVER (Yates and Etzioni 2009) (RS), and DIRT (Lin and Pantel 2001). USP and OntoUSP first parse input text using the Stanford dependency parser (de Marneffe, MacCartney, and Manning 2006), learn an MLN for semantic parsing from the dependency trees, and output this MLN and the MAP semantic parses of the input sentences. These parses formed the knowledge base (KB). To answer questions, the semantic parses of the questions (with the question slot replaced by a dummy word) are matched to parses in the KB via subsumption test. In addition, when OntoUSP matches a question to its KB, it not only considers the lambda-form cluster of the question relation, but also all its sub-clusters.

Table 1 shows the results comparing OntoUSP with other systems. USP easily dominates all other systems except for OntoUSP in both precision and recall. In particular, its accuracy is 9 points higher than the second best and it extracted more than three times as many correct answers as the second best. OntoUSP further substantially improved on the recall of USP by 47%, while incurring no loss in precision. Compared to TextRunner, OntoUSP extracted more than five times of correct answers.

Manual inspection shows that both USP and OntoUSP resolve many nontrivial syntactic variations without user supervision. They consistently resolve the syntactic difference between active and passive voices and identify many distinct argument forms that mean the same (e.g., "X stimulates Y" ≈ "Y is stimulated with X", "expression of X" ≈ "X expression"). They also resolve many synonymous expressions for entities and relations. Their large performance gain over previous systems illustrate the advantage of applying Markov logic and joint inference.

Additionally, OntoUSP gains on recall over USP by the induced ISA hierarchy. Like USP, OntoUSP discovered the following clusters (in core forms) that represent some of the core concepts in biomedical research:

{regulate, control, govern, modulate}
{induce, enhance, trigger, augment, up-regulate}

{inhibit, block, suppress, prevent, abolish, abrogate, down-regulate}

However, USP formed these as separate clusters, whereas OntoUSP also induces ISA relations from the INDUCE and INHIBIT clusters to the REGULATE cluster. This allows OntoUSP to answer many more questions that are asked about general regulation events, while the text states them in specific directions. Below is an example question-answer pair output by OntoUSP; neither USP nor any other system were able to extract the necessary knowledge.

Q: What does IL-2 control?

A: The DEX-mediated IkappaBalpha induction.

Sentence: Interestingly, the DEX-mediated IkappaBalpha induction was completely inhibited by IL-2, but not IL-4, in Th1 cells, while the reverse profile was seen in Th2 cells.

This illustrates the importance of discovering ISA relations and performing hierarchical smoothing.

Conclusion

Machine reading has attracted increasing interest in recent years. A breakthrough in it will result in profound impacts in many fields. In this paper, we analyze the core challenges in machine reading and propose a unifying approach based on statistical relational AI. We also present OntoUSP, an end-to-end machine reading system based on Markov logic and joint inference, as well as the experimental results in a machine reading task. The dramatic performance gain of OntoUSP over previous state-of-the-art systems demonstrates the promise of our approach and highlight the significance of statistical relational AI.

Key directions for future work include: identify bootstrap sources with easily extractable knowledge, develop a unifying learning framework to incorporate various forms of direct or indirect supervision, scale up joint inference with coarse-to-fine inference, induce probabilistic ontology for efficient coarse-to-fine inference, develop continuous learning systems for community creation, etc.

Acknowledgements

We thank the anonymous reviewers for helpful comments. This research was partly funded by ARO grant W911NF-08-1-0242, AFRL contract FA8750-09-C-0181, DARPA contracts FA8750-05-2-0283, FA8750-07-D-0185, HR0011-06-C-0025, HR0011-07-C-0060 and NBCH-D030010, NSF grants IIS-0534881 and IIS-0803481, and ONR grant N00014-08-1-0670. 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, NSF, ONR, or the United States Government.

References

Bakir, G.; Hofmann, T.; B. Schölkopf, B.; Smola, A.; Taskar, B.; Vishwanathan, S.; and (eds.). 2007. *Predicting Structured Data*. Cambridge, MA: MIT Press.

Banko, M.; Cafarella, M. J.; Soderland, S.; Broadhead, M.; and Etzioni, O. 2007. Open information extraction from the web. In *Proc. of IJCAI*.

Basu, S., and Kapoor, A. 2009. *Proceedings of the NIPS-09 Workshop on Interactive Machine Learning*. MIT Press.

Cimiano, P. 2006. *Ontology learning and population from text*. Springer.

de Marneffe, M.-C.; MacCartney, B.; and Manning, C. D. 2006. Generating typed dependency parses from phrase structure parses. In *Proc. of the Fifth International Conference on Language Resources and Evaluation*.

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

Downey, D.; Etzioni, O.; and Soderland, S. 2010. Analysis of a probabilistic model of redundancy in unsupervised information extraction. *Artificial Intelligence*.

Dudik, M.; Blei, D.; and Schapire, R. 2007. Hierarchical maximum entropy density estimation. In *Proc. of ICML*.

Felzenszwalb, P. F., and McAllester, D. 2007. The generalized A* architecture. *Journal of Artificial Intelligence Research* 29.

Fellbaum, C., ed. 1998. *WordNet: An Electronic Lexical Database*. Cambridge, MA: MIT Press.

Gelman, A., and Hill, J. 2006. *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Cambridge University Press.

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

Hearst, M. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proc. of COLING*.

Kiddon, C., and Domingos, P. 2010. Ontological lifted probabilistic inference. In *submission*.

Kim, J.-D.; Ohta, T.; Tateisi, Y.; and Tsujii, J. 2003. GENIA corpus - a semantically annotated corpus for bio-textmining. *Bioinformatics* 19:180–82.

Lin, D., and Pantel, P. 2001. DIRT - discovery of inference rules from text. In *Proc. of KDD*.

McCallum, A.; Rosenfeld, R.; Mitchell, T.; and Ng, A. 1998. Improving text classification by shrinkage in a hierarchy of classes. In *Proc. of ICML*.

Petrov, S. 2009. *Coarse-to-Fine Natural Language Processing*. Ph.D. Dissertation, Dept. of Comp. Sci., Univ. of Berkeley.

Poon, H., and Domingos, P. 2008. Joint unsupervised coreference resolution with Markov logic. In *Proc. of EMNLP*.

Poon, H., and Domingos, P. 2009. Unsupervised semantic parsing. In *Proc. of EMNLP*.

Poon, H., and Domingos, P. 2010. Unsupervised ontological induction from text. In *Proc. of ACL*.

Poon, H., and Vanderwende, L. 2010. Joint inference for knowledge extraction from biomedical literature. In *Proc. of NAACL*.

Snow, R.; Jurafsky, D.; and Ng, A. 2006. Semantic taxonomy induction from heterogenous evidence. In *Proc. of COLING/ACL*.

Staab, S., and Studer, R. 2004. *Handbook on ontologies*. Springer.

Suchanek, F.; Kasneci, G.; and Weikum, G. 2008. Yago - a large ontology from Wikipedia and WordNet. *J. of Web Semantics*.

Suchanek, F.; Sozio, M.; and Weikum, G. 2009. Sofie: A self-organizing framework for information extraction. In *Proc. of WWW*.

Wu, F., and Weld, D. S. 2008. Automatically refining the wikipedia infobox ontology. In *Proc. of WWW*.

Yates, A., and Etzioni, O. 2009. Unsupervised methods for determining object and relation synonyms on the web. *JAIR*.