Review of Jan Dědek's PhD thesis entitled "Semantic Annotations"

Despite its overly general title, the thesis is really an exploration of how inductive logic programming and ontologies can help the task of information extraction from unstructured texts. The latter task is extremely important for the realization of the semantic web and it is a well motivated plan to explore the two evidently relevant technologies in its context. I would have appreciated a more explicit statement of the main encompassing goal of the thesis, however, specific low-level goals are formulated at each subproblem, and they work out the puzzle in a logical way.

Contributions

The main purpose of ILP in the thesis is to learn extraction rules mapping the syntactic structure of text tokens into categories of meaning (such as detection of specific kinds of events). There has been some previous work on using ILP for natural language processing, but what distinguishes the present thesis therefrom, and what I very much like about it, is that it blends ILP into the most advanced NLP framework. This mainly means the integration of ILP into the state of the art tool GATE, the exploitation of deep linguistic parsing including the concept of tectogrammatical dependencies for the construction of training data, and the conversion of learned rules into a unified ontological representation. This is a nice practical contribution which pushes ILP's state of the art in a direction largely unattended by ILP researchers. The impact of this contribution is reduced by the inferior extraction accuracy of the ILP-based methods w.r.t. the perceptron learning method built in the GATE tool. So the contribution is rather a proof of concept than a best practice. I believe though that future work could lead to significant improvements of the performance of the explored method; I am providing some hints later in this review.
A theoretical contribution of the thesis is represented by the concept of extraction ontologies. Being no ontologist, I cannot deeply evaluate this concept but by common sense I consider it smart indeed to develop a method that can retain extraction rules in a way that is independent of the underlying learning tool and that preserves the semantics of the rules, i.e. allows to act upon them.
In my own perception, the last key contribution is the experimental evaluation of the fuzzy ILP framework (originally proposed by the thesis advisor) for prediction of accident severity from the unstructured text. Fuzzy ILP achieved very promising results in this task.
The thesis also provides a methodology for manual design of extraction rules, which I consider less interesting than the contributions above.
Summing up contribution-wise, the thesis is of sufficient novelty and quality and thus ripe for a defense.

Formal aspects

I liked a lot that the text is narrated as a story of problem-solving, which makes it well understandable for the reader. However, some lines of reasoning which would better be presented compactly are broken down into different chapters and connected by cross-references. Apart from minor grammar flaws and a some funny tautologies ("document annotation is a term, which usually refers to the process of putting annotations to documents", "the idea of shareable extraction ontologies assumes that extraction ontologies will be shareable..."), the language reads well and is unambiguous. A good point is that generally, the author does not pretend that simple things are complicated by introducing overly complicated formalism, which is quite a common flaw of dissertations. On the other hand, some parts would indeed deserve a formally more precise exposition (see comments/questions below).

Recommendation

The author has demonstrated his potential for independent scientific work and I recommend his thesis for a defense. I have the following specific comments/questions which the author may want to address on that occasion.

Specific comments/questions

1. In 4.7.1.,.2 : it is not clear what really is B, in particular whether it is a set of only ground facts or it allows more general clauses. This becomes confusing with Def 3 where \mathcal{B} is defined as a mapping from B (which would standardly be a set of clauses) whereas in the text it is used such that it assigns a number to a *predicate*. What is probably really meant is *ground facts;* but then Def 3 should not speak of predicates but (ground) facts.
2. Is the problem of classification of accident severity really relational? All ground facts seem to refer to the primary entity (represented by FOL variable A) so they look very much like attributes of instances just as in propositional learning. The problem could still be relational if there were multiple facts of a given predicate associated with one entity, e.g. injury(A, light), injury(A, serious) for two injuries in a single accident. But is that the case?
3. I was not happy with the simple conclusion that ILP does not perform very well in the extraction-rule learning task. First of all, a deeper investigation should be done to determine *why* the performance was not good. I would start this "debugging" by replacing the perceptron-based competitor with a symbolic (interpretable) classifier, such as decision trees. If the latter also outperforms ILP, one could rather easily explore in it what kind of decision-conditions make the alternative classifier accurate and why the current language bias used in the ILP method is not able to formulate them. There are many possible sources of the low performance, for example the way numeric attributes are handled. For example the learned condition *damage_atl(A,1500000)* does not seem to make too much sense since it clearly overfits to the one specific value. Were predicates such as greater_than(X,Y) available to the learner at all? Note that Aleph provides the *lazy evaluation* method for dealing with numbers. The *tDependency* predicate is often employed in the learned rules in a chain-like manner. Would it help to define its transitive closure in the background knowledge? Finally, Aleph is a general purpose system with a large set of parameters (proof depth, clause length, search method, …), which influence the final performance. Did you tune their best values through internal validation? Did you try to change them at all? Also, due to its general applicability, Aleph is known to be slow so your runtime findings are not surprising. But since your learning data are trees, you could have taken advantage of much faster ILP methods appropriate for this kind of data. For example, the system ReLF (Kuzelka & Zelezny, Machine Leaning 2011) is especially suited for tree-like relational structures and works faster by orders of magnitude. It could be useful also in the fuzzy ILP setting especially given that multiple runs of conventional ILP are exectuted per each fuzzy ILP experiment.

August 19, 2012

doc. Ing. Filip Železný, Ph.D
Dpt. of Cybernetics
Faculty of Electrical Engineering
Czech Technical University in Prague