Learning Information Extraction Rules: An Inductive Logic Programming approach

James Stuart Aitken[1]

Abstract.
The objective of this work is to learn information extraction rules by applying Inductive Logic Programming (ILP) techniques to natural language data. The approach is ontology-based, which means that the extraction rules conclude with specific ontology relations that characterise the meaning of sentences in the text. An existing ILP system, FOIL, is used to learn attribute-value relations. This enables instances of these relations to be identified in the text. In specific, we explore the linguistic preprocessing of the data, the use of background knowledge in the learning process, and the practical considerations of applying a supervised learning approach to rule induction, i.e. in terms of the human effort in creating the data set, and in the inherent biases in the use of small data sets.

1 Introduction

Automatically deriving a semantic interpretation of free text is a challenging research task [11, 12] which has an immediate and pressing application in improving access to the large volumes of knowledge published on-line. The relevance of ILP to extracting a machine-processable semantic representation from free text, e.g. MEDLINE abstracts [3] and database queries [11], and from semi-structured webpages [4, 7] has been demonstrated. Applications range from more intelligent information retrieval, to the construction of knowledge bases and knowledge discovery [3]. To this list we add the task of marking-up web documents with ground relation instances (RDF triples) that is needed for the Semantic Web and the intelligent tools it promises.

This paper explores the problem of learning information extraction rules that accurately derive ground facts characterising the content of natural language texts. We use the FOIL ILP learner[13], and therefore the problem becomes one of constructing the appropriate representation of the text, and of the background knowledge that is available. Naturally, this must be done automatically. The relations that are learned are those defined in a pre-existing ontology of the domain. These relations cover the fraction of the content of the texts that we are interested in.

The domain is that of chemical compounds and other concepts related to *global warming*, and we are interested in assertions as to the level of, and changes in emission rates and concentrations of greenhouse gases. The ontology provides the classes and the subclass relations, and defines the attributive relations such as *emission rate*, and *concentration level*. The texts used in the experiments reported here are taken from the popular science publications *Nature* and *New Scientist*.

[1] Artificial Intelligence Applications Institute, Division of Informatics, University of Edinburgh, Edinburgh, EH1 1HN, Scotland.

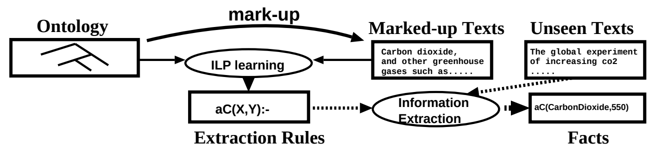

Figure 1. Rule learning and IE

The techniques presented here are not domain dependent. However they are dependent on the availability of resources such as an ontology, and the capability to recognise named entities. There may be some dependence on the text genre.

Rule induction is viewed as part of a semi-automated process which necessarily includes human involvement. A domain expert, and/or a knowledge engineer, may need to understand the suggested extraction rules. They may also wish to refine the suggested rules. Alternatively, the rules can be learned and applied incrementally, with the human having the role of correcting the derived facts/mark-up once an initial set of rules is learned. Human involvement is also required for creating a small data set of ontologically marked-up texts which is used by the induction algorithm. In addition to finding techniques to represent textual data such that good rules are learned, we also consider the costs and complexities of human involvement in the process. Figure 1 shows the processes of mark-up, rule learning and information extraction.

Inductive Logic Programming is appropriate for this learning task for several reasons: ILP provides a natural representation of the relations to be learned (including type restrictions); the framework allows alternative sentence representations to be explored; and background knowledge can easily be represented [11, 12]. The FOIL algorithm seeks to specialise rules and uses an information-gain heuristic to provide a measure of the coverage of a clause. This measure allows for noise in the data, correlates with precision and recall, and provides a uniform metric to assess alternate sentence representations and background theories.

The global warming domain, and ontology that was created to represent it are described in Section 2. Section 3 begins by describing the text representation input to FOIL, and considers the background theories made available to the learning algorithm. The evaluation of the ILP-based rule learning approach is presented in Section 4, and this is followed by an evaluation of the same task when carried out by a knowledge engineer in Section 5. Related work is discussed in Section 6, and, finally, we draw some conclusions.

2 Ontology Development and Text Mark-up

The texts on global warming were retrieved by keyword search from *Nature* and *New Scientist* articles. The keywords used include *co2*,

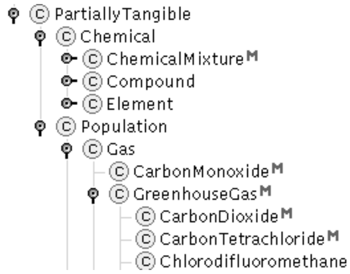

Figure 2. Excerpt of the domain ontology

oxygen, *greenhouse gas*, and *global*, in various combinations. Not all texts are about global warming, one is about the irrigation of greenhouses with sea water, others are about coal production and the use of methane as a fuel. An ontology was created to represent this domain.

The global warming domain ontology is a synthesis of several existing ontologies, since none could be found that covered all of the relevant aspects of this domain: the Ontolingua ontology of chemical elements [6], the categorisation of chemical compounds found in EcoCyc [9], and the proposals of [8] distinguishing heterogeneous and homogeneous populations. Combining these ontologies was a purely manual process which also took additional sources of domain knowledge into account. Figure 2 shows a small excerpt of the ontology.

The predicates representing the quantitative and qualitative relationships of interest are: 1) *atmosphericConcentration* (*aC*) holding of a *Gas* and a real number; 2) *atmosphericConcentration_Qual* (*aCQ*) holding of a *Gas* and one of the symbols $\{high, low\}$; 3) *changeInAtmosphericConcentration* (*cAC*) holding of a *Gas* and one of the symbols $\{increase, decrease, none\}$; 4) *changeInRateOfEmission* holding of a *Gas* and one of the symbols $\{increase, decrease, none\}$; 5) *causes* holding of an *Event* or a *PartiallyTangible* and an *Event*; and 6) *stateOfMatter* holding of solid, liquid and gaseous (these concepts are modelled as attribute values).

In marking-up sentences with these six relations, it was decided to ignore the temporal context in which the assertion holds. Several texts refer to past eras, e.g. the Miocene period, and report inferred greenhouse conditions at that time. A complete description of the content of a text would represent such contextual information. However, that is beyond the scope of this work. The mark-up we construct is simply the central description:
`cAC(CarbonDioxide,increase)` whether that be a past, present or hypothesised statement, Figure 3 shows more examples. Such assertions are contained in XML terms which are embedded within the sentences they describe. The XML can be eliminated from the texts for further NLP processing, or can be extracted when required.

In this manner, sentences in the text set are manually marked-up with ontology terms. This mark-up is later extracted from the text to form the target relations which are required during supervised learning. These terms also form the set of valid statements that can be made about the texts during the testing procedure. Consequently, it is important that the texts are completely and correctly marked-up in order to provide accurate inputs to learning and testing. The effort and expertise required in this task, and the repeatability and consistency of this procedure are therefore important issues. Marking-up the 30 texts (containing 205 sentences, 5862 words) used in this experiment took approximately 4 hours, including the time for typing in the XML annotations and reviewing the outcome. No tools were available to assist this task, beyond a standard text editor.

The global experiment of increasing atmospheric CO2 concentrations by burning fossil fuels has neither a control nor replicates
`<target name="cAC(CarbonDioxide,increase)"/>`.
So it is difficult to quantify how much faster the world's forests might be growing under high CO2 conditions
`<target name="aCQ(CarbonDioxide,high)"/>`.
Higher levels of CO2 can clearly make plants grow better
`<target name="aCQ(CarbonDioxide,high)"/>`.

Figure 3. Text from Nature [5] with annotation

As a test of the consistency of mark-up, a set of 6 randomly selected texts were double-marked by a second knowledge engineer. First and second markers found 46 and 54 relations respectively, in these texts. In 44 cases the same mark-up was made. It was noted that there are few instances of co-reference to be resolved in determining the arguments of attribute relations. This may be a general feature of the popular science genre. In conclusion, agreement between markers was greater than 90% once errors and differences in assumptions of scope are removed.

3 Inputs to Learning

The input to an ILP algorithm consists of two parts: the list of positive and negative instances of the relation for which rules should be learned (the target relation), and, secondly, the background theory from which the rules should be constructed. The input to FOIL has a third component, the type definition, which we discuss below.

The target relations which are input to FOIL are simply extracted from the XML terms in the marked-up texts. All that is required in addition is to index these relations with a sentence identifier. In this application of ILP, the background theory has two components, one derived from the the texts and one from the semantic theory, and we now discuss these elements in turn.

3.1 Sentence Representation

NLP techniques can be used to enrich the information given to the machine learning algorithm, or to filter the input. For example, part of speech tags may be included in the sentence-word relation, and may also be used as a filtering mechanism, e.g. words marked as determiners may be removed from the learning input.

The following techniques may (optionally) be employed in the NL processing stage of our system:

- Part of speech (POS) tagging: The Brill tagger [1] is used.
- Morphological analysis: Words are stemmed by the morphological analyser of [10].
- POS tag convergence: The Brill tags for each major category are replaced by a single tag for each type (i.e. by one tag for all six types of noun).
- POS filtering: The POS tags are used to exclude certain categories of word.
- Frequency analysis: The frequency of occurrence of each word across the text set is measured. Low frequency word can be filtered.
- Named-Entity Recognition: The ontology concepts, or instances of ontology concepts, found by named-entity recognition are added to the sentence representation.
- Context: The immediate context of certain words is found. Currently this is simply the immediate successor of words which are potential attribute values.

The *hasWord(Sentence-ID,POS,WORD)* relation represents the tokens in a sentence, and their part of speech. The options listed above determine whether the word is stemmed, and whether the POS tag is modified from the original. Named entities are added to the sentence by the same relation (the special tag *ne* is introduced): *hasWord(Sentence-ID,ne,NamedEntity)*. In the domain we are address, the named entity is typically an ontology class. Context information is represented by the relation *context(Sentence-ID, Word-1, Word-2)* where Word-2 is the context of Word-1. This extends the sentence representation with a structural relation between tokens.

3.2 Ontology Theory

The *isa(Class,Class)* relation of the ontology can be used as a background theory. Only a subset of these relations need be included - those which define the class hierarchy between a concept identified in the target relation and the type of the relation. From these it should be possible to learn generalisations. Instances of classes can also be treated.

The second resource that can be used is a mapping between a concept and the form of words that it may be associated with in a text. For example, the concept *CarbonDioxide* has two *txtform* relations:

```
txtform1(CarbonDioxide,co2).
txtform2(CarbonDioxide,carbon,dioxide).
```

This is a resource that may be created manually, or may be learned. In fact, we can use the marked-up data set to suggest correlations between concepts and word forms, but we do not explore that in depth here. The *txtform1(Class,Word)* and *txtform2(Class,Word,Word)* relations contain similar information to that used in named-entity recognition. These relations explicitly provide the mapping information to the learner, while named-entity recognition results in a *hasWord* relation stating that a concept occurs in a specific sentence.

3.3 Data Types and Data Set Biases

The FOIL algorithm requires that the target relation and the background theory relations be typed, and that types be extensionally defined in the input file. At a minimum, the type of a relation must include all terms that appear in the tuples. Where the types of a relation are specified in the ontology we can use the ontology definition to identify and construct FOIL types. When this is not the case (as for *hasWord*), or the set of terms is too great (e.g. the integers) the FOIL type must cover all tuples plus an approximation of the list of constants that may occur in that type.

As we are dealing with natural language data we need to ensure that rules are not over-generalised due to an over-constrained definition of the allowable arguments of relations. On the other hand, if we were to use only a single type for all relations which contained all words (which may number several thousand) then the search space would be prohibitively large.

We have developed the following heuristics for FOIL type definitions. The cases to consider are where the arguments of the ontology relation are an (abstract) ontology class (e.g. the first argument of *aC*); numerical (e.g. the second argument of *aC*); and symbolic (e.g. the second argument of *aCQ*).

- For ontology types: the FOIL type includes the names of all ontology classes below the class defining the type of relation (e.g. all classes below *Gas* for *aC*).
- For symbol types: the FOIL type includes all symbols available in the symbol set (e.g. the set *increase, decrease, none* for *cAC* - whether or not these occur in the target set).
- For numerical types: the FOIL type should include all numbers in target relation, plus all numbers that occur in the sentences for which there are target relations.

As noted above, rule learning is necessarily performed on a relatively small set of marked-up texts. A larger set would provide more representative data, but the cost in human labour would be prohibitive. As a consequence, the data set is biased by the initial selection criteria. While not all texts are about about global warming, there are clearly unrepresentatively high correlations between words and concepts in the target relation such as *increase* and *CarbonDioxide*. In addition, there is a lack of examples of the co-occurrence of these terms in a sentence which is not described by the target relation. Constructing additional negative target relations would not address this problem, as the issue is the unrepresentative nature of the language data presented to the learner. Adding a sentence where the highly correlated concepts occur in a sense which is not described by the target relation would help induce rules which better distinguish positive and negative cases. Thus, a set of bias *hasWord* relations is constructed by generating a new sentence ID and stating that this sentence contains *increase* and *CarbonDioxide*. That is, we assume it is plausible that the concepts co-occur in some sentence, without actually finding a real example sentence.

FOIL has a closed-world flag (`-n`), and does not require explicitly-constructed negative examples—two features which are very useful when learning from natural language data where negative examples do not naturally arise.

4 Evaluation of ILP Rule Learning

The processing techniques listed above are evaluated on the test data. First the experimental methodology is stated.

4.1 Experimental Method

The sentences in the text-base are randomly divided into training and testing sets in a 2/3 to 1/3 split. Ten such random data sets are created and each experiment is run 10 times. Each experiment attempts to learn rules for three ontology relations: *atmosphericConcentration* (*aC*), *atmosphericConcentration_Qual* (*aCQ*) and *changeInAtmosphericConcentration* (*cAC*). The average number of relations of each type in the training and testing sets is given in Table 1. To show the extent of the relations used in the experiments, Table 2 lists the average number of ground relations used — in addition to those which represent the text (the Baseline) for *cAC*.

Average	Sentences	aC	aCQ	cAC
Training	137	8.5	7.5	16.0
Testing	68	2.5	3.5	7.0

Table 1. Number of sentences and relations

Exp.	Predicate	No.	Exp.	Predicate	No.
Baseline	*hasWord*	288.4	Ontology	*isa*	9.0
txtform	*txtform1/2*	10.0	Bias	*hasWord*	9.0
ne	*hasWord ne*	93.6	Context	*context*	14.7

Table 2. Number of ground predicates

The standard performance measures of precision, recall and F score are calculated as follows. Precision is the ratio of derived relations which are correct to the total number of derived relations. Recall is the ratio of the number of correct relations that can be derived to the total number of correct relations. The F score is calculated giving equal weight to precision and recall. The average performance in a test for each measure is quoted. As the F score is calculated for each trial in an experiment, then averaged, it will be used as a summary score where appropriate. The set of correct relations for any sentence is just those in the XML mark-up. The set of derived relations is all relations that can be inferred from the learned rules.

4.2 Results

The first results include the baseline performance of the learner, plus the F scores obtained by adding each of the five knowledge sources to the learning input individually: the text mapping predicate, named-entities, the ontology relation (*isa*), the bias relations, and the context assertions. Having documented this information, combinations of sources are examined.

Exp.	aC	aCQ	cAC	Exp.	aC	aCQ	cAC
Baseline	0.19	0.68	0.41	Ontology	0.19	0.63	0.40
txtform	0.19	0.62	0.38	Bias	0.00	0.51	0.40
ne	0.19	0.57	0.40	Context	0.41	0.59	0.45

Table 3. Baseline results (F score)

Table 3 shows the F score for *aC* (the numerical relation) to be low (0.19) in all but the experiment where context is added. This is due to over-specialisation as the rules contain specific numbers. However, when bias information is added the rules identify the unit of measure, `ppmv`, of the number which is in fact a qualitative improvement. The addition of context information gives a more promising F score for *aC*, derived from rules which have a useful degree of generalisation. The second set of experiments examine combinations of knowledge sources.

The following combinations are explored, all include bias and context information: **A** baseline; **B** text mapping; **C** named-entity; **D** ontology; **E** text mapping and named-entity; **F** text mapping and ontology; **G** named-entity and ontology; and finally, **H** text mapping, named-entity and ontology.

Exp	aC			aCQ			cAC		
	P	R	F	P	R	F	P	R	F
A	1.00	0.59	0.62	0.62	0.67	0.59	0.58	0.48	0.47
B	1.00	0.59	0.62	0.69	0.67	0.64	0.67	0.51	0.55
C	1.00	0.55	0.57	0.65	0.73	0.57	0.47	0.56	0.47
D	1.00	0.59	0.62	0.69	0.67	0.64	0.66	0.46	0.48
E	1.00	0.55	0.57	0.94	0.73	0.76	0.57	0.45	0.47
F	1.00	0.59	0.62	0.69	0.67	0.64	0.62	0.46	0.47
G	1.00	0.59	0.62	0.73	0.67	0.66	0.67	0.42	0.50
H	1.00	0.59	0.62	0.89	0.67	0.68	0.70	0.45	0.54

Table 4. Results of combination tests (**P**recision, **R**ecall and **F** score)

The new baseline system, including both bias and context, enables *aC* relations to be learned - note the increase in F score of 21% over the previous best score. This is because the association between numbers and their unit of measure is now represented. The *context* relation is used widely in rules for all target relations. Other knowledge sources do not greatly alter performance on the *aC* relation.

The precision scores for *aCQ* and *cAC* can be improved by 32% and 12% respectively by the various combinations of knowledge sources. Similar improvements in recall scores are not achieved. Adding a knowledge source does not necessarily improve performance. Test **C** shows that adding named entities improves recall at the expense of precision and F score. However, the combination of named entities with other information is beneficial as subsequent tests show. The changes in performance are not necessarily repeated across the relations. Test **H** shows the best combined performance across the relations: precision ≥ 0.70, F ≥ 0.54.

The following are examples of the rules learned:

```
B: cAC(A,'Methane',increase):-
      hasWord(A,v,increase),
      txtform1('Methane',E),hasWord(A,F,E).

C: aCQ(A,'CarbonDioxide',C):-
      hasWord(A,D,'CarbonDioxide'),hasWord(A,E,C).

F: cAC(A,B,increase):-
      hasWord(A,v,increase),isa(B,E),
      txtform2(E,F,G), hasWord(A,H,G).

H: cAC(A,B,increase):-
      hasWord(A,D,B),hasWord(A,v,rise).
```

The **B** rule says that *Methane* increases if the word `increase` and the text form of *Methane* occur in a sentence. This shows the intended use of the *txtform* relation. **C** states that *CarbonDioxide* has concentration C (high or low) if both the named entity `CarbonDioxide` and C occur in a sentence. This shows generalisation over the symbol set. Rule **F** is an interesting over-generalisation, saying that there in an increase in B if E is a superclass of B, and E increases. **H** states that there is an increase in concentration of a named entity B if that entity occurs in a sentence along with the verb `rise`. This is an example of generalisation over ontology classes, plus the identification of a term with similar semantics to the symbol *increase*.

Less encouragingly, rules often identified words with no semantic relation to the relation being described. This is not surprising given that these words are currently only selected on the basis of information gain. Further, due to the bias in the data, a test for an occurrence of a concept was often missed, e.g.

`cAC(A,'CarbonDioxide',C):-context(A,C,atmospheric).`

This rule fails to test for any reference to *CarbonDioxide* in sentence A. It is sufficiently accurate when applied to both the training and testing sets as they have the same biases.

4.3 Analysis

Given the small numbers of examples we work with it appears unlikely that generalisations of the grammatical structure of the training sentences could be learned. The use of the bag-of-words representation allows a scaling down of the size of the training data set: The text representations employed capture just enough information for the attribute relations to be learned. There may be genre-specific factors at work as writing for popular science requires clarity, hence complex relative clauses may be less frequent, and, as noted, there are few anaphoric references to the main topics of a sentence.

To address concerns over the small size of the data sets, we repeated three experiments with an enlarged data set of 371 sentences. There was no consistent pattern of change in the scores. However, the variance in F score across the 10 trials was consistently reduced: The enlarged data set improves the estimate of the F score by reducing the 95% confidence interval.

The experiments addressed learning attribution relations, and in these cases the directionality of the relation is unambiguous. Clearly there are limits to this approach. It would work in some common-sense cases, to quote a familiar example: *The parliament was bombed by the guerillas*, but not in others *Microsoft sues IBM*. Knowledge of the subject and object in the sentence will certainly be required to construct the correct relation in the second example. We look to incorporate the structure of phrases into the sentence representation in future work.

The known bias in the learning data did affect the rules learned in that certain assumptions that hold of the data were exploited. One solution is to manually review the rules in order to achieve similar performance on unseen texts. We next present some initial results on the complexity of manually authoring and reviewing rules.

5 Evaluation of Manual Rule Authoring

The tasks of rule authoring and rule revision were given to a knowledge engineer to compare performance and get a qualitative insight into the complexities of the tasks.

A knowledge engineer was provided with a listing of the texts, with and without mark-up, the ontology was described as was the procedure for rule evaluation. In the first task, the aim was to write rules to generate the given mark-up. No computer support was available. The second task was to refine the rules and for this automated scoring on the training set was provided. The rule sets were saved at each stage of the experiment and the times to complete the tasks were recorded. In order to make the task as natural as possible, no stemming or other processing of the texts was done.

The knowledge engineer spent 28 minutes reading the texts and 90 minutes writing rules. The F scores on the testing data for the first task are: *aC:* 0.67 *aCQ:* 0.57 *cAC:* 0.00. After revision the score for *cAC* was 0.60, no revisions to the rule sets for *aC* or *aCQ* were made so the scores are unchanged. The revision process took 45 min.

Against expectation, the majority of the effort was spent on the initial analysis phase. This produced very good rules. Indeed, few changes were made during revision. However, a critical change to a *cAC* improved performance considerably. It was found particularly difficult to cover the negative cases (sentences for which no relation should be deduced) and to know which concepts were of most value to cover (by their frequency of occurrence) by inspection. The rules were longer than automatically learned rules, typically containing 4 conjuncts. The coverage of ontology concepts was greater than actually occurred in the texts, and therefore the rules are more widely applicable. It was noted that the task was a very unnatural one, despite the intention to reduce the programming element. The lack of a *context* relation was noted. This exercise confirmed the belief that an initial mark-up of the texts is necessary, and would have been required had it not already been provided. One hour and 58 minutes was required to construct the first rule sets. FOIL usually takes less than 10 seconds to produce a rule set. When all knowledge sources are available to FOIL the rule quality is comparable, and therefore we conclude that editing the automatically generated rules is an efficient alternative to extensive manual analysis.

6 Related Work

The features used for sentence representation in ILP approaches to information extraction [3, 7] and text classification [2, 4] have strong similarities. Token-based featured such as the *has_word* predicate which denotes the occurrence of word in a web page [4] are often used in combination with structural features such as *next_token* [7] or *next_phrase* [3]. The ability to represent relations between tokens, as opposed to purely propositional features, is a strong argument in favour of the ILP approach (although for text classification, the experimental evidence for a performance benefit is not overwhelming [2]). The *hasWord* and *context* relations in the present work are the analogues of the token-based and relational features cited.

The information extraction problem is often framed as identifying a sequence of tokens that fill a slot in a template. For this reason the induced rules may take the *features* themselves as arguments, e.g. the *every(Feature,Value)* predicate used in SRV [7]. The IE problem we address induces rules which conclude with a terms in a predefined symbol set, the ontology. Symbols such as ontology classes never occur in the text, but their presence may be indicated by named-entity recognition when that module is applied. Attribute values may correspond with tokens in the text, however. This division of named-entity recognition from IE is beneficial as sequence identification is removed from the learning task. Structural relationships between tokens can be learned from the sentence encoding directly.

7 Conclusions

For the purpose of extracting attribute relations from text we have shown that a bag-of-words sentence representation, combined with a simple word context relation is adequate. Attribute relations are common in scientific ontologies and we have shown how extracting these relations can exploit the typing of the ontology.

The inclusion of grammatical information in sentence representation is typically assumed; however, we have shown that it is possible to achieve success without it. Learning from small training sets becomes a possibility as complex structures need not be learned. This is not to ignore the need for deeper analysis: We envisage a two stage IE process where the grammatical information is acquired on-demand, for those sentences whose interpretation requires it.

Acknowledgements

This work is supported under the Advanced Knowledge Technologies (AKT) Interdisciplinary Research Collaboration (IRC), which is sponsored by the UK Engineering and Physical Sciences Research Council under grant number GR/N15764/01.

REFERENCES

[1] Brill E. A simple rule-based part-of-speech tagger. *Proc. of ANLP-92, 3rd Conference on Applied Natural Language Processing* 1992, pp. 152–155.

[2] Cohen, W. Learning to classify English text with ILP methods. In *Advances in Inductive Logic Programming* ed. De Raedt, L., 1995, pp. 3-24.

[3] Craven, M. and Kumlien, J. Constructing biological knowledge bases by extracting information from text sources. *Proc. 7th International Conference on Intelligent Systems for Molecular Biology* 1999.

[4] Craven, M. and Slattery, S., Relational learning with statistical predicate invention: Better models for hypertext. *Machine Learning* 43(1-2), 2001, pp. 97-119.

[5] Davidson, E.A. and Hirsch, A.I. Carbon cycle: Fertile forest experiments. *Nature* 411, 24 May 2001, pp. 431-433.

[6] Fernández López, M., Gómez-Pérez, A., and Pazos Sierra, J. Building a Chemical Ontology using Methontology and the Ontology Design Environment. IEEE Intelligent Systems 14:1, Jan-Feb, 1999, pp. 37–46.

[7] Frietag, D. Information extraction from HTML: Application of a general machine learning approach. *Proc. AAAI 98*, 1998, :517-523.

[8] Hafner, C.D and Fridman, N. Ontological foundations for biology knowledge models. *Proc. 4th International Conference on Intelligent Systems for Molecular Biology*, St. Louis, AAAI Press, pp. 78-87.

[9] Karp, P.D. and Riley, M. EcoCyc: The resource and lessons learned. in ed. Letovsky, S. *Bioinformatics* Kluwer, 1999.

[10] Minnen, G, Carroll, J. and Pearce, D. Robust, Applied Morphological Generation. *Proc. First International Natural Language Generation Conference* 2000, pp. 201-208.

[11] Mooney, R.J. Learning for semantic interpretation: scaling up without dumbing down. *Proc. Workshop on Learning Language in Logic* Springer Verlag, 1999.

[12] Muggleton, S. Inductive logic programming: issues, results and the LLL challenge. *Artificial Intelligence* 114(1-2), 1999, pp. 283-296.

[13] Quinlan, R.J. and Cameron-Jones, R.M. FOIL: A midterm report. *Proc. European Conference on Machine Learning* 1993.