Learning from Natural Language Data using ILP: The Role of Background Knowledge and Negative Examples

Stuart Aitken and Stephen Potter

AIAI, University of Edinburgh
stuart,stephenp@aiai.ed.ac.uk,
http://www.aiai.ed.ac.uk

Abstract. This papers explores the task of learning a partial semantic interpretation of natural language texts using inductive logic programming (ILP). The requirements for learning adequate rules using two different ILP algorithms, GOLEM and FOIL, are determined and compared. The formulation of positive and negative examples of sentences and the representation of background knowledge is explored. Our findings confirm the necessity to bias the search for rules in a domain where only small numbers of positive examples occur naturally. We present heuristics for formulating background knowledge and negative examples and in the NLP domain.

1 Introduction

Learning from natural language data has become an important application of ILP techniques. The tasks addressed vary from learning parsing rules [16], to information extraction (IE) from text [3] and from the web [5, 9]. The specific nature of the data - sentences and their structure - has even motivated the development of new ILP algorithms, e.g. FOIDL [13]. In the field of information extraction, learning rules for template filling has been recognised as a means of replacing, or augmenting, the rule authoring process. Active learning [17] and the combination of automated learning with interactive (human) correction have been proposed [4] in recognition that user interaction and user-provided feedback are important features of a usable learning system.

The learning algorithms in IE systems are typically specialised to the IE task in hand. For example, specific representations of pattern-matching templates, or of the generalisation procedure may be used. This can make drawing general conclusions about these learning problems a difficult task. We are interested in identifying these properties.

Given an input text, information extraction aims to instantiate a template which represents the important relationships between the facts in the text. These techniques are important in dealing with the large amounts of textual material in company intranets, and on the World Wide Web. The proposed Semantic Web requires a vast amount of mark-up, hence the automated extraction of instances

of ontology relations from text will be a key challenge. The formulation of the IE task that we adopt is exactly that required for automating mark-up: An ontology of a given domain is assumed to exist, and we aim to learn rules that derive ground instances of ontology relations from text. A small initial set of marked-up texts is created from which to learn the mark-up rules.

In this paper, we apply two existing ILP systems to the information extraction task (as defined above). The algorithms have different search strategies and heuristics. By applying the algorithms to the same task we aim to get an insight into the general properties of the language learning problem that make it a complex task for an ILP system, without exploiting any particular feature of one system. Similar heuristics have been shown to be successful when used in FOIL [1]. Rather than re-implementing the two algorithms, we modify the data which is input to the algorithms. This may suggest beneficial changes to the algorithms - as other authors have found. We propose and test a number of techniques for data representation whose effect is to direct ILP learning towards better (higher precision and recall) rules. The evaluation is empirical, and we also give a qualitative assessment of the rules learned.

The following section, Section 2 presents the ILP systems we use. The domain and data set is explained in Section 3, and our ideas on negative examples are explained in Section 4. The heuristics we develop and the method of the study are described in Section 5, and the results are presented in Section 6. The relationship of these results to existing work and conclusions are drawn in Sections 7 and 8.

2 FOIL and GOLEM

ILP systems attempt to learn logic programs from a combination of training examples and background knowledge. Each positive training example represents an instantiation of the conclusion of the target rule to be learned, while the (optional) negative examples represent instantiations of the conclusion that are inconsistent with this rule.

FOIL [15] induces rules by a top-down covering algorithm which starts with an empty body and attempts to add the best clause to it. Those positive examples that are covered by the new rules are removed from the set of positive examples, to get a new training set, and the procedure is repeated. An information-gain heuristic is used to determine the best clause. This measure is a function of the positive and negative tuples covered.

Learning in GOLEM [14] is founded on the notion of *least general generalisation* (lgg). The lgg of two logical clauses is a clause which, through the conservative introduction variables in place of terms, subsumes both clauses in a cautious generalisation of the two. Extending this idea, the *relative least general generalisation* (rlgg) of two clauses is their lgg relative to some shared background knowledge.

When presented with a learning task, then, GOLEM selects a number of pairs of positive examples at random and computes their rlggs with respect to the supplied background knowledge. Of these, the rlgg that covers the greatest

number of positive examples, and the fewest negative, is selected, and adopted as the working hypothesis for the target rule. The positive examples covered are removed from consideration, and the procedure repeats.

In this manner, GOLEM learns in a 'bottom-up' fashion: positive examples are used to successively raise the level of generalisation of hypotheses. This might be contrasted with a top-down algorithm such as FOIL which initially proposes a very general hypothesis and subsequently lowers its level of generality as learning progresses. This difference is reflected in each algorithm's use of negative examples. In FOIL, the negative examples — whether provided explicitly or generated internally — constrain the choice of appropriate specialisations. GOLEM, on the other hand, does not require negative examples for learning, but if they are available, they are used to guide learning by checking the consistency of hypotheses.

3 Domain and Data

Extracting information from technical publications and scientific journals in the life sciences is our ultimate aim. The results can be used in a number of ways: from the improved indexing of documents, to the creation of knowledge bases representing the facts in an abstract [6]. The search tools typically used by biologists are keyword based, and have the usual associated problems. An ontology-based characterisation of an abstract would complement the increasing acceptance of the need for ontologies in biology, for example, in the annotation of gene products [10]. As a step towards this goal, we address the problem of extracting information from the journal *Nature* on the topic of global warming. Before describing the learning task we introduce the ontology that was created for this domain, and the text mark-up that was performed to create the initial data set of semantically annotated texts.

The global warming domain ontology is a combination 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 [8], and the proposals of [11]. Combining these ontologies was a purely manual process.

The predicates representing the quantitative and qualitative relationships of interest are: 1) *atmosphericConcentration* (aC) holding of a *Gas* and an integer; 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 one of the attributes solid, liquid and gaseous.

3.1 Text Mark-up

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,

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)"/>`.
But will Earth's vegetation absorb from the atmosphere, and retain, much of the CO2 pouring out of our exhaust pipes and smoke stacks? If it does, then the threat of global warming from increasing CO2 would be less severe
`<target name="cAC(CarbonDioxide,increase)"/>`
`<target name="causes(CarbonDioxide,GlobalWarming)"/>`.

Fig. 1. Text from Nature [7] with annotation

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. Such assertions are contained in XML terms which are embedded within the sentences they describe. Figure 1 shows more examples. The XML can be eliminated from the texts for further NLP processing, or can be extracted when required.

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. They 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.

4 Negative Examples and Bias Sentences

ILP learning is guided by the negative examples of the target relation. In the context of natural language data, the negative examples are the false interpretations of all sentences in the input data. Bearing in mind that only true interpretations are contained in the mark-up, the complete set of false interpretations is all other ontologically-valid statements that can be generated.

The negative examples are used to evaluate quality of a rule and/or of the addition of a clause to a rule. This may be through a coverage measure i.e. number of examples which are explained, coverage viewed as information gain, or probabilistic estimate depending on the algorithm. A simple enumeration of all false relations is not always a practical way to proceed as the algorithms fail (or the available implementations fail). The aim is to find a set E_-, for all target relations T holding of $A\{a_1...a_n\}$, $B\{b_1...b_n\}$ and $C\{c_1...c_n\}$ from the sets E_+

(also composed triples of the same sets) and BG, the background knowledge, that adequately characterise the false interpretations. A is the set of sentence numbers, and B and C are equivalent to the respective ontology classes for the target relation.

This problem is further complicated by the nature of the source data from which the rules are to be learned: the training data may only be a partial representation of the potential range of inputs.

In addition, despite the inclusion in the set of training texts of a number of texts which are not semantically related to the main topic, but are only related by word occurrence, there remains a disproportionately high correlation between concepts in the domain. (I.e. the conditional probabilities of the co-occurrence of words in a sentence are not representative of texts as a whole.)

The problem is that there are no sentences in the language data in BG where the concepts occur independently of one another, i.e. when T holds of $< a_i b_j c_k >$ there is typically no *hasWord* relation holding of any sentence a in which a word indicating b_j occurs without the occurrence of the word indicating c_k. In the case of target relations which hold of numbers (e.g. let C be the set of real numbers), and assuming c_k occurs in a sentence, the situation is usually that c_k never occurs again in the *hasWord* relation.

While we can specify E_- to exclude $T(\_, \_, c_k)$ this will not prevent overgeneralisation as there is no background *hasWord* instance to which any hypothesised rule might apply — except the one instance $hasWord(a_i, d_y, c_k)$ which is the evidence for the positive target.

The solution is to construct a set of *hasWord* tuples where all elements of C in E_+ occur, but no element of B occurs. This 'bias' sentence has the interpretation false for all B, thus E_- can be extended with these tuples. The bias tuples replace the need for a completely representative set of input texts - which would be in infeasible proposition.

The intended effect of the combining negative examples with bias relations is to force search away from exploiting the inherent non-representativeness in the language data, and towards finding rules which include the words which occur in sentences and which distinguish the interpretations. The success of this will depend on an adequate sentence representation, i.e. one which captures distinguishing features which occur sufficiently often to be predictive. In addition to the *hasWord* relation, a *context* relation is introduced to denote that, in a given sentence, *word-1* occurs in the context of *word-2*. In common with *hasWord*, the *context* relation may be over generalised in learning and so we investigate the interaction of bias with this relation.

We now describe the implementation of this theory in two ILP learners. In each case, the input to the learner is modified, and the inputs to each learner are as similar as possible. In all cases, the set of positive examples of the target relation is derived from the marked-up sentences.

4.1 FOIL

FOIL requires all relations to be explicitly typed, and for the types to be enumerated. The definition of these types is important as they act as a constraint in the search for rules. While FOIL can learn from positive examples only, we shall explicitly provide negative examples. This is done for consistency with the experiments for the other algorithms. The inputs to FOIL are constructed as follows:

- The types *sent*, *pos*, *word*, *arg1*, *arg2* for sentence numbers, parts of speech, words, and the first and second arguments of the target relation respectively, are declared. The types are constructed as follows:
  - 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.
  - The types for ontology, numerical, symbol, words and parts of speech are all constants, the sentence types (denoting sentence numbers) are not.
- The set of negative examples is the cross product of sentence numbers, target relation argument-1 and argument-2, minus the positive examples.
- In addition, when a context relation is used in conjunction with a bias sentence, each word in the bias sentence has a context assertion. A don't-care value is the unspecified context.

4.2 GOLEM

GOLEM does not require the types of relations to be explicitly specified. However, that information was provided by unary relations in order to provide information consistent with that provided to FOIL. Constants need not be declared in GOLEM, and the modes (usage of arguments as inputs/outputs) of relations were not specified although this is possible.

- The relations *sent*, *pos*, *word*, *arg1*, *arg2* for sentence numbers, parts of speech, words, and the first and second arguments of the target relation respectively, are added to the background theory. They are constructed as for FOIL.
- The set of negative examples is the permutation of each sentence number for which there is a target and the argument-1 value for that target relation, with some argument-2 value (selected such that values not previously used in the negative examples are preferred). A second negative example is constructed for each sentence using an argument-1 value which does not occur in the positive target set. This results in a small but representative set of negative examples.

- When a bias sentence is used, a set of negative target relations, as defined above, is stated to hold of the bias sentence.
- In addition, when a context relation is used in conjunction with a bias sentence, each word in the bias sentence has a context assertion. The context is a arbitrary symbol (a gensym). These gensysms are defined to be words, and to occur in the bias sentence. This mirrors the don't-care symbol used in FOIL.

The set of negative examples is a small subset of those given to FOIL. This is due the different role played by these examples in evaluation in GOLEM. With a noise parameter of 0, the extent of the negative examples is not critical, the important factor is that they prohibit the over-generalisation of the target relation by covering the possible cases that may arise. For this reason the negative examples include not only permutations of the argument-1 and 2 values, but also use argument-1 values which do not occur in the observed positive examples.

5 Method

The experiments aim to confirm the postulated theory that a combination of negative examples and bias relations are required in order to learn IE rules. We are also interested to discover whether the background theories we believe to be important improve performance (these are described below). To demonstrate this, we test each ILP system with one of a number of background theories added to the sentence representation. We also run a baseline test with only the sentence data, and a combination test where both bias and context tuples are added to the background theory. Unlike the other background theories, the context relation is derived from the sentence data, and hence may require the specialisation which is believed to be caused by the bias relations. The context relation was previously shown to be essential to learning numerical attribute relations.

The experiments do not evaluate fully-optimised IE systems, but rather the results will show the potential for building IE systems using the methods we describe.

The representation of sentences, and of the background information provided to the ILP algorithms is described below. The same representation was used in all trials. The design of the trials is then outlined.

5.1 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.

Through experiment, the following techniques were found to be beneficial in preparing the text data:

- Part of speech (POS) tagging: The Brill tagger [2] is used.
- Morphological analysis: Words are stemmed by the morphological analyser of [12].
- 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, for example, numbers are excluded when learning a non-numeric relation.
- Frequency analysis: The frequency of occurrence of each word across the text set is measured. Words occurring less than three times across the text set are filtered out.
- 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 their immediate successor in the sentence.

The basic sentence representation is a set of tuples <Sentence-ID,POS,Word>. These are denoted by the *hasWord* relation. 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)*. 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 beyond the basic bag-of-words.

5.2 Experimental Method

The 205 sentences in the test set 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 carried out with a 10-fold cross-validation. Each experiment attempts to learn rules for three ontology relations: *atmosphericConcentration* (*aC*), *atmosphericConcentration_Qual* (*aCQ*) and *changeInAtmosphericConcentration* (*cAC*). The first is a numerical attribute, the second and third are symbolic attributes. Each of these relations takes an ontology class as the first argument. The average number of relations of each type in the training sets (each containing 137 sentences) is: *aC:* 8.5 *aCQ:* 7.5 *cAC:* 16.0. In the testing sets (each 68 sentences): *aC:* 2.5 *aCQ:* 3.5 *cAC:* 7.0.

The standard performance measures of precision, recall and F score are used. 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 a test, then averaged, it is the most indicative measure.

Exp	aC			aCQ			cAC		
	P	R	F	P	R	F	P	R	F
Baseline	0.05	1.00	0.10	0.72	0.73	0.70	0.00	1.00	0.01
Text mapping	0.05	1.00	0.10	0.72	0.73	0.70	0.00	1.00	0.01
Named-entity	0.05	1.00	0.10	0.72	0.73	0.70	0.00	1.00	0.01
Ontology	0.05	1.00	0.10	0.72	0.73	0.70	0.00	1.00	0.01
Bias	1.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00	0.01
Context	0.06	1.00	0.11	0.74	0.67	0.67	0.00	1.00	0.01
Bias+Context	0.90	0.47	0.51	1.00	0.18	0.25	0.00	1.00	0.01

Table 1. Results of GOLEM tests (**P**recision, **R**ecall and **F** score)

6 Results

The effect of adding each individual element of the background theory in turn is assessed, namely: the text mapping predicate, named-entities, the ontology relation, the bias relations, and the context assertions. This is repeated for each ILP system.

6.1 Results for GOLEM and FOIL

The results for GOLEM are given in Table 1. In the four experiments: Baseline, text mapping, named-entity and ontology, GOLEM learns rules which contain one clause, typically:
`aC(S,'CarbonDioxide',B):- hasWord(S,cd,B).`
For the *aCQ* relation, the POS argument in *hasWord* is that for adjectives, as opposed to numbers.
`aC(S,'CarbonDioxide',B):- hasWord(S,j,B).`
None of the other background theories are made use of. The higher scores achieved for the *aCQ* rules are a result of the scoring procedure which counts only relations that are valid in the ontology. This eliminates many of the adjectives, but few of numbers to which the bodes of these rule apply to.

When used alone, the bias relation causes over-fitting for *aC* and *aCQ* relations. Specific numbers occur in the *aC* rules. Adding the *context* relation should permit *aC* to be learned, but the results show rules to be overly general. Combining bias and context causes high precision and recall rules to be learned for *aC*, F=0.51. These rules test for numbers which occur in the context of `ppmv`, ppmv being a unit of measure for gases. The *context* relation is also useful in learning *aCQ*, but this time adding bias causes a marked reduction in recall, such that F score is much reduced. For *cAC*, GOLEM learns rules which are over-general:
`cAC(S,A,increase).` despite there being information in the background theory that can identify `A` with named-entities in the sentence.

Table 2 contains the results for FOIL on the same tests. For the *aC* relation in tests Baseline, text mapping, named-entity and ontology, FOIL learns rules

Exp	aC			aCQ			cAC		
	P	R	F	P	R	F	P	R	F
Baseline	0.52	0.50	0.04	0.70	0.70	0.66	0.48	0.50	0.47
Text mapping	0.52	0.50	0.04	0.77	0.73	0.73	0.64	0.43	0.47
Named-entity	0.52	0.50	0.04	0.65	0.73	0.57	0.59	0.40	0.44
Ontology	0.52	0.50	0.04	0.79	0.63	0.65	0.52	0.47	0.44
Bias	0.80	0.00	0.00	0.80	0.59	0.58	0.49	0.50	0.48
Context	0.45	0.72	0.34	0.69	0.67	0.64	0.49	0.43	0.42
Bias+Context	0.45	0.82	0.45	0.69	0.67	0.64	0.50	0.44	0.43

Table 2. Results of FOIL tests (**P**recision, **R**ecall and **F** score)

which are over-fitted, in that specific numbers are contained in the rules:
`aC(S,'CarbonDioxide',550):- hasWord(S,_,550).`
When bias is added, the rules are further specialised by adding tests for the occurrence of `part` or `ppmv` to the rule. In this case, recall is reduced to 0. When *context* is added, this information is used to construct rules which are adequately general. When bias and context are combined, recall is improved and F score improves to 0.45.

For the aCQ relation, FOIL derives rules which test for specific words in sentences, similar to the test for numbers in aC. FOIL makes use of the named-entity information when that is provided, but does not use the ontology theory. The bias information increases specialisation, and causes specialisation in the use of context, but the effect is not reflected in the quantitative results.

FOIL is able to learn rules for cAC, and use text mapping, named-entity and ontology theories, with the effect of improving precision (but not F score) over the baseline. The use of context and bias do not produce notable changes.

6.2 Comparison

GOLEM over-generalises rules for aC while the FOIL rules tend to be over-fitted. Only context and bias significantly affect the rules learned for these relations. GOLEM never uses three of the five theories, while for aCQ and cAC FOIL does make use of them. Adequate rules for these relations are learned by FOIL, with the alternate theories affecting the quality of the rules to a relatively small extent. GOLEM is seen to over-generalise rules for cAC in all experiments. The contrasting behaviour of FOIL and GOLEM on cAC can be attributed to the explicit typing in FOIL. The argument-1 and 2 types are declared to include all possible values. For GOLEM this declaration is just another unary predicate in the background theory, while in FOIL it is a statement of the extent of the relation. The extent is known from the ontology, and may be wider than is actually observed in the target relations. For example, for some data sets cAC holds only of *CarbonDioxide* and consequently there are no targets to be covered for other values. However, negative examples are constructed for all possible values, and FOIL induces rules which exclude these. GOLEM is not forced to

generalise over an enumerated type as FOIL is, but instead over-generalises based on the more limited range of observed tuples: from *CarbonDioxide* to $\forall x$.

7 Related Work

The construction of knowledge bases from MEDLINE abstracts described in [6] is very similar to the IE task we address. An existing ontology of classes and relations is assumed, and the objective is to identify the relations from the text. However, Craven and Cumlien initially construct Bayesian models which relate sentences to their semantic interpretation. This is extended to combine a relational approach, using a FOIL-like algorithm, with the Naive Bayes classifier. A direct comparison of results is complicated by the use of different data sets and measures of performance. At 90% precision the recall is 20% (approximately) for the relational classifier. These figures, and the corresponding F score 33%, are comparable with our results.

We have discussed the need for negative examples, and this issue has also arisen in the context of learning the past tense of English verbs. A method for calculating the implicit coverage of negative examples by a clause is proposed in [13]. The ability to estimate this value replaces the need for an explicit enumeration of negative examples. The estimate is $u^v - p$, where u is the number of words/constants available, v is the number of free variables in the clause for which we want to know the negative coverage, and p is the number of positive examples which unify with the clause. This estimate is modified if a variable is partially instantiated, e.g. [a,c,t|X]. This estimate is implemented in the FOIDL algorithm. Our approach is less direct, we do not modify the coverage or information-gain measures, but explicitly construct the inputs to which they are applied. The motivation is similar, being the lack of explicit negative targets in natural language data. Our approach is applicable to rlgg-based systems, and accounts for other properties of the data.

8 Conclusions

We have shown that with the appropriate treatment of negative examples, and the use of bias relations, it is possible to learn information extraction rules using ILP. The approach we have adopted does not use highly-engineered representations for sentences, in contrast, we attempt to redress two important properties of the source data: the lack of negative examples, and the unrepresentative nature of any small sample of texts which leads to a high correlation of the terms we are interested in. The evaluations are comparative, and do not indicate any limit of the level of performance that may potentially be achieved using ILP. Better performance has previously been reported using more optimised settings for FOIL [1].

The use of bias relations is shown to be essential for learning relations which take numerical values, as otherwise the rules for these relations are over-

generalised. Bias relations can cause over-fitting in other cases. We find this effect across two ILP learners.

Our experiments also show that simply providing background theories as input to learning may not have the intended effect: GOLEM makes no use of three of the five theories in any experiment. The explicit enumeration of types that FOIL requires is seen to be beneficial, particularly where the language data does not reflect the range of terms that may potentially be seen.

References

1. Aitken, S. Learning Information Extraction Rules:An Inductive Logic Programming approach. *Technical Report*, Division of Informatics, University of Edinburgh, 2002.
2. 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.
3. Califf, M. E. and Mooney, R. J. Relational Learning of Pattern-Match Rules for Information Extraction. *Proc. AAAI 1999*, pp. 328-334.
4. Ciravegna, F. and Petrelli, D. User Involvement in Adaptive Information Extraction. *IJCAI-2001 Workshop on Adaptive Text Extraction and Mining* 2001
5. Craven, M.,DiPasquo, D., Freitag, D., McCallum, A., Mitchell, T., Nigam, K., and Slattery, S., Learning to extract symbolic knowledge from the World Wide Web. *Proc. AAAI* 1998, pp. 509-516.
6. 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.
7. Davidson, E.A. and Hirsch, A.I. Carbon cycle: Fertile forest experiments. *Nature* 411, 24 May 2001, pp. 431 - 433.
8. 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.
9. Frietag, D. Information extraction from HTML: Application of a general machine learning approach. *Proc. AAAI 98*, 1998, :517-523.
10. The Gene Ontology Consortium
`http://www.geneontology.org/`
11. 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.
12. Minnen, G, Carroll, J. and Pearce, D. Robust, Applied Morphological Generation. *Proc. First International Natural Language Generation Conference* 2000, pp. 201-208.
13. Mooney, R.J. Induction of First-Order Decision Lists: Results on Learning of the Past Tense of English Verbs. *JAIR*, 3 (1995) pp: 1-24.
14. Muggleton, S. and Feng, C. Efficient Induction in Logic Programs. In Muggleton, S. editor, *Inductive Logic Programming*, Academic Press, 1992, pp. 281-298
15. Quinlan, R.J. and Cameron-Jones, R.M. FOIL: A midterm report. *Proc. European Conference on Machine Learning* 1993.
16. Tang, L.R. and Mooney, R.J. Using multiple clause constructors in inductive logic programming for semantic parsing. *Proc. 12th European Conference on Machine Learning* 2001.
17. Thompson, C. A., Califf, M. E., and Mooney, R. J. Active Learning for Natural Language Parsing and Information Extraction. *Proceedings of the Sixteenth International Machine Learning Conference (ICML-99)* 1999, pp. 406-414.