IJCAI-95 Workshop on New Approaches to Learning for Natural Language Processing

Issues in Inductive Learning of Domain-Specific Text Extraction Rules*

Stephen Soderland, David Fisher,
Jonathan Aseltine, Wendy Lehnert
Department of Computer Science
University of Massachusetts, Amherst, MA 01003-4610
{soderlan dfisher aseltine lehnert}@cs.umass.edu

Abstract

Domain-specific text analysis requires a dictionary of linguistic patterns that identify references to relevant information in a text. This paper describes CRYSTAL, a fully automated tool that induces such a dictionary of text extraction rules. We discuss some key issues in developing an automatic dictionary induction system, using CRYSTAL as a concrete example. CRYSTAL derives text extraction rules from training instances and generalizes each rule as far as possible, testing the accuracy of each proposed rule on the training corpus. An error tolerance parameter allows CRYSTAL to manipulate a trade-off between recall and precision. We discuss issues involved with creating training data, defining a domain ontology, and allowing a flexible and expressive representation while designing a search control mechanism that avoids intractability.

1 Domain-specific Text Analysis

Considerable domain knowledge is needed by a system that analyzes unrestricted text and identifies information relevant to a particular domain. The text analysis system must consider domain-specific vocabulary and semantics, as well as linguistic patterns typically used in references to domain objects. A system tailored to extracting information from newswire stories about terrorism might identify the subject of the passive verb "kidnapped" as the victim of a kidnapping event. On the other hand, if the system is to analyze medical patient records, it will need a totally different set of rules that identify references to various types of diagnoses and symptoms.

The University of Massachusetts BADGER text analysis system instantiates a set of case frames called "concept nodes" (CN's) to represent information extracted from a document, using a dictionary of rules called CN definitions. BADGER performs selective concept extraction similar to that of its predecessor, the CIRCUS system [Lehnert, 1991; Lehnert *et al.*, 1993].

Creating CN definitions by hand is a laborious task, which requires both knowledge of the domain and a deep understanding of the text analysis system. Researchers at UMass spent about 1500 hours creating a dictionary of CN definitions for the ARPA-sponsored 1991 MUC-3 evaluation, whose domain was Latin American terrorism. The following year a partially automated dictionary construction tool, AutoSlog [Riloff, 1993], was developed. AutoSlog automatically created a set of proposed CN definitions which required less than 8 hours of review by a "human in the loop". The AutoSlog CN dictionary acheived 98% of the performance of the hand-crafted dictionary.

The CRYSTAL system presented in this paper [Soderland *et al.*, 1995], carries the idea of AutoSlog further. It creates a dictionary of CN definitions with greater expressiveness that that of AutoSlog, and is fully automatic.

2 Concept Node Definitions

A CN definition specifies a set of syntactic and semantic constraints that must be satisfied for the definition to apply to a segment of text. Before the CN definition is applied, BADGER must segment the input text and identify the syntactic constituents of each segment such as subject, verb phrase, direct and indirect object, and prepositional phrases. BADGER looks up the semantic class of each word in a domain-specific semantic lexicon and instantiates a concept node with the extracted information if all constraints are met.

Examples in this paper are from a medical domain, where the task is to analyze hospital discharge reports, and identify references to "diagnosis" and to "sign or symptom". These are further classified with subtypes.

Diagnosis:	Sign or Symptom:
confirmed	present
ruled out	absent
suspected	presumed
pre-existing	unknown
past	history

*This research was supported by NSF Grant no. EEC-9209623, State/ Industry/ University Cooperative Research on Intelligent Information Retrieval. Thanks also to David Aronow for help as our medical domain expert.

The example shown in Figure 1 is a CN definition from this domain that identifies references to absent symptoms. This CN definition extracts the phrase in the direct object buffer when the subject buffer has the semantic class <patient>, the verb is "denies" in the active voice, and the direct object has the semantic class <sign_or_symptom>.

```
CN-type: sign_or_symptom
Subtype: absent
Extract from Direct Object
Active voice verb
Subject constraints:
    head class <patient>
Verb constraints:
    words include "DENIES"
Direct Object constraints:
    head class <sign_or_symptom>
```

Figure 1: A CN definition to identify "sign_or_symptom, absent"

This CN definition would extract "any episodes of nausea" from the sentence "The patient denies any episodes of nausea". It would fail to apply to the sentence "The patient denies a history of asthma", since asthma is of semantic class <disease_or_syndrome>, which is not a subclass of <sign_or_symptom>.

In this paper we will describe how CRYSTAL induces a set of CN definitions from training instances. CRYSTAL will be used as a concrete example to ground a discussion of various aspects of using machine learning to generate text extraction rules. We will discuss preparation of a training corpus, definition of a domain ontology, representation of training instances and extraction rules, and designing search control to handle an intractable problem efficiently.

3 Overview of Dictionary Induction

CRYSTAL derives a domain-specific dictionary of text extraction rules from a training corpus, initializing the dictionary with a CN definition for each positive training instance. These initial CN definitions are designed to extract the relevant phrase in the training instance that motivated them, but are too specific to apply broadly to previously unseen sentences.

Figure 2 shows the initial CN definition derived from the sentence fragment "Unremarkable with the exception of mild shortness of breath and chronically swollen ankles." The domain expert has marked "shortness of breath" and "swollen ankles" with CN type "sign_or_symptom" and subtype "present". When BADGER analyzes this sentence, it assigns the complex noun phrase "the exception of mild shortness of breath and chronically swollen ankles" to a single prepositional phrase buffer. When a complex noun phrase has multiple head nouns or multiple modifiers, the class constraint becomes a conjunctive constraint. Class constraints on words such as "unremarkable" that are of class <root_class> are dropped as vacuous.

```
CN-type: sign_or_symptom
Subtype: present
Extract from Prepositional Phrase "WITH"
Verb = <NULL>
Subject constraints:
    words include "UNREMARKABLE"
Prepositional Phrase constraints:
    preposition = "WITH"
    words include "THE EXCEPTION OF MILD
                   SHORTNESS OF BREATH
                AND CHRONICALLY SWOLLEN
                ANKLES"
    head class <sign_or_symptom>,
               <physiologic_function>,
               <body_part>
    modifier class <sign_or_symptom>
```

Figure 2: An initial CN definition, including all exact words and classes

Before the induction process begins, CRYSTAL cannot predict which characteristics of an instance are essential to the CN definition and which are merely accidental features. So CRYSTAL encodes all the details of the text segment as constraints on the initial CN definition, requiring the exact sequence of words and the exact sets of semantic classes in each syntactic buffer.

The main work of CRYSTAL is to gradually relax the constraints on these initial definitions to broaden their coverage, while merging similar definitions to form a more compact dictionary. Semantic constraints are relaxed by moving up the semantic hierarchy or by dropping the constraint. Exact word constraints are relaxed by dropping all but a subsequence of the words or dropping the constraint. The combinatorics on ways to relax constraints becomes overwhelming. There are over 57,000 possible generalizations of the initial CN definition in Figure 2.

The goal of generalizing is to drop constraints derived from accidental features of the motivating instance while retaining constraints that identify positive instances and constraints that eliminate negative instances. Relaxing constraints enough to unify a CN definition with a similar CN definition tends to accomplish these goals.

CRYSTAL finds useful generalizations of an initial CN definition, D, by locating a highly similar CN definition, D'. A new definition, U, is then created with constraints relaxed just enough to unify D and D'. This involves dropping constraints that the two do not have in common and finding a common ancestor of their semantic constraints. The new CN definition, U, is then tested against the training corpus to make sure that it does not extract phrases that were not marked with the CN type and subtype being learned.

If U is a valid CN definition, CRYSTAL deletes from the dictionary all definitions covered by U, thus reducing the size of the dictionary while still covering all the positive training instances. In particular, D and D' will be deleted. Then U becomes the current CN definition and this process is repeated, using similar CN definitions

The CRYSTAL algorithm:

```
Initialize Dictionary and Training Instances Database
Do until no more initial CN definitions in Dictionary
    D = an initial CN definition from Dictionary
    Loop
        D' = the most similar CN definition to D
        If D' = NULL, exit loop
        U = the unification of D and D'
        Test the coverage of U in Training Instances
        If the error rate of U > Tolerance
                exit loop
        Delete all CN definitions covered by U
        Set D = U
    Add D to the Dictionary
Return the Dictionary
```

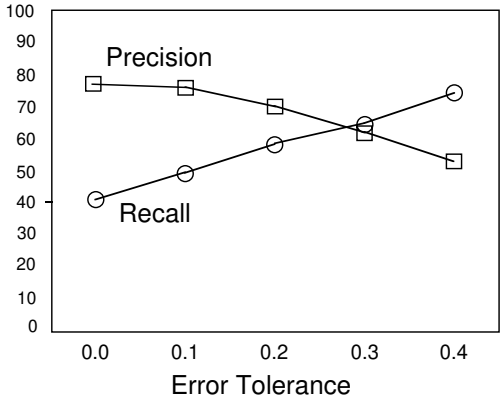

Figure 3: Effect of the error tolerance setting on performance

to guide the further relaxation of constraints. Eventually a point is reached where further relaxation would produce a CN definition that exceeds some pre-specified error tolerance. At that point, CRYSTAL begins the generalization process on another initial CN definition until all initial definitions have been considered for generalization.

CRYSTAL's learning algorithm is similar to the inductive concept learning described by Mitchell [1982] and that described by Michalski [1983]. Each uses a data-driven search to find the most general rule that covers positive training instances without covering negative instances. CRYSTAL is more robust than Mitchell's version space algorithm, which assumes no noise in the training and searches for a single concept to cover all positive instances. Michalski sees his approach in terms of a minimum covering set problem, where the goal is a minimal set of generalizations that cover all the positive instances. This is exactly the goal of CRYSTAL.

4 Experimental Results

Experiments were conducted with 337 hospital discharge reports, with a total of 13,000 instances, of which 5,100 were "diagnosis" and 1,900 were "sign or symptom". These were partitioned into a training set and a blind test set, then dictionaries of CN definitions were induced from the training set and evaluated on the test set.

Performance is measured here in terms of recall and precision, where recall is the percentage of possible phrases that the dictionary extracts and precision is the percentage correct of the extracted phrases. For instance if there are 50 phrases that could possibly be extracted from the test set by a dictionary, but the dictionary extracts only 20 of them, recall is 40%. If the dictionary extracts 25 phrases, only 20 of them correct, precision is 80%.

Thesetting of an error tolerance parameter can be used to manipulate a trade-off between recall and precision of a dictionary. Figure 3 shows performance of a dictionary of CN definitions that identify "sign_or_symptom" of any subtype, where the error tolerance is varied from 0.0 to 0.4. The results shown here are the averages of 50 random partitions of the corpus into 90% training and 10% test documents for each error tolerance.

To assess the learning curve as training size increases, we chose the partition size for training to be 10%, 30%, 50%, 70%, and 90%. This was done 50 times for each training size and results averaged as before. The number of positive training instances in a partition depends on what CN type and subtype is being learned. Figure 4 shows recall for the most frequent CN type and subtypes as the number of positive training instances increases. The error tolerance is set at .20 for this experiment.

These results are for CN definitions acting in isolation, which explains the different coverage of the subtype "absent" and "present" for "sign_or_symptom". CRYSTAL is able to add a constraint requiring the word "no" for absent symptoms, but has no mechanism to require the absence of "no" for present symptoms. Additional noun phrase analysis rules are needed to handle conjunction, disjunction and the scoping of negation. Future work will address these issues by moving to a granularity of simple noun phrase within syntactic buffers and scoping for negation before applying CRYSTAL's CN definitions.

5 Issues in Automatic Dictionary Induction

With CRYSTAL serving as a concrete example, we now turn to a discussion of some of the design and implementation issues involved in automatic dictionary induction.

5.1 Preparing a Training Corpus

CRYSTAL uses a supervised learning approach to induce rules that identify relevant information in the text. This requires a set of positive and negative training instances of the information to be extracted.

As will often be the case, the text analysis task was initially ill-defined for the hospital discharge report domain, and our first challenge was to specify a taxonomy of what information was to be considered relevant in this domain. We worked closely with a physician to defined a taxonomy of subtypes of "diagnosis" and of "sign or symptom".

Annotation of training texts was done by the physician and three nurses, using a point-and-click interface that

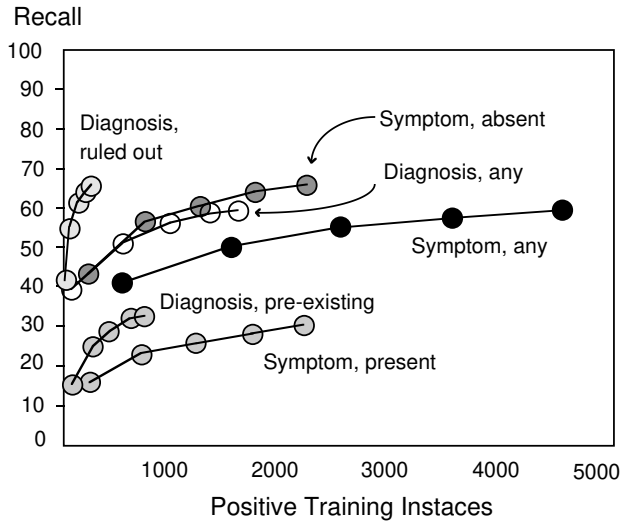

Figure 4: Learning curve: recall increases with the number of training instances

inserted SGML style tags around phrases to label them with the appropriate CN type and subtype. As actual tagging began, the task specification was further refined through discussion with the physician who served as our domain expert. The general guidelines were to tag simple phrases, but deciding exactly which phrase to label was not always straightforward.

The sentence "Abdomen was swollen but nontender" is tagged as follows to indicate that "swollen" has CN type "sign or symptom" with subtype "present" and that "nontender" has CN type "sign or symptom" with subtype "absent".

Abdomen was <SP> swollen </SP> but
<SA> nontender </SA> .

In sentences where the descriptive phrase is essentially empty of content, such as "unremarkable" or "normal", we debated what phrase should be tagged. In such cases the body part itself is tagged as sign or symptom, absent. An example of this is "Abdomen was unremarkable", which is annotated as follows.

<SA> Abdomen </SA> was unremarkable.

Consistency in training is essential if a supervised learning algorithm is to succeed in finding compact representations of the rules to be learned. In particular, a phrase that is accidentally missed during tagging will become a negative training instance. Coding guidelines and careful quality control are essential.

5.2 Creating a Domain Ontology

Precisely defining a taxonomy of relevant information is only part of creating a domain ontology. CRYSTAL also requires a semantic hierarchy and a semantic lexicon that maps terms into semantic classes in that hierarchy. Early in CRYSTAL's development, we created our own ad hoc semantic hierarchy and semantic lexicon with some assistance from our domain expert. This was replaced by one derived from the Unified Medical Language Systems (UMLS) on-line medical MetaThesaurus and Semantic Network [Lindberg *et al.*, 1993]. UMLS is under developement by the National Library of Medicine and represents a knowledge source far superior to our ad hoc efforts, but is itself under development and is still weak in coverage of terms for clinical findings.

Our ad hoc semantic hierarchy grouped together semantic classes that occur in similar contexts for our task. For example the words "patient", "woman", "female", and "she" all belong to the semantic class <human> in the ad hoc hierarchy. In the UMLS hierarchy these have widely scattered semantic classes. A "patient" is of class <patient_or_disabled_group> which is a descendant of <idea_or_concept>. A "woman" is of class <human> which is a <physical_object>. A "female" is of class <organism_attribute> which is an <idea_or_concept>, and "she" is not even listed in the semantic lexicon.

UMLS also places the class <sign_or_symptom> in a subtree so far removed from <disease_or_syndrome> that their only common ancestor is the root of the entire hierarchy. Our ad hoc hierarchy gave them a common ancestor that at least distinguished them from non-medical terms.

Some rules that can be expressed compactly with our ad hoc semantic hierarchy require a disjunction or several rules using UMLS. We would expect CRYSTAL to require a larger training set when using UMLS than with the task-specific hierarchy. On the other hand, with 134 classes as opposed to 27 in the ad hoc hierarchy, UMLS could be expected to support finer grained distinctions.

For training sizes up to 50 documents, a CN dictionary based on UMLS lagged behind the task-specific ontology in coverage. Figure 5 compares recall of CN dictionaries based on UMLS and on the ad hoc ontology. Both are dictionaries that identify CN type "sign_or_symptom" of any subtype, using error tolerance of 0.20.

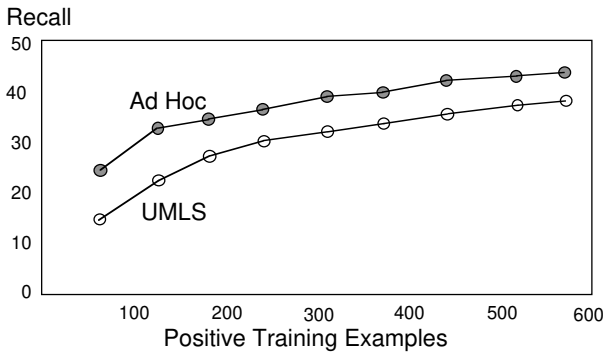

Figure 5: Comparing recall with UMLS and with an ad hoc ontology

5.3 Expressiveness of Extraction Rules

The representation of training instances and rules is a fundamental design decision, particularly when dealing with a complex phenomenon such as natural language. A richly expressive representation will often cause an unacceptable increase in computational complexity, while

a simpler representation glosses over important distinctions.

One issue is the scope of the "local context" to be included in text extraction rules. Shall we include all phrases in the same clause as potentially important to the context or exclude some parts of the clause to make processing more tractable? Another issue is the granularity of representation. Should the representation include exact words, or the semantic classes of words, or a combination of words and classes?

The CN definitions that CRYSTAL generates may include constraints on any or all syntactic buffers in a text segment as analyzed by BADGER. A segment is generally a simple clause with buffers such as subject, verb phrase, direct or indirect object, and prepositional phrases. Each buffer may be constrained to include a sequence of specific words, to have specific semantic classes in the head noun, or to have specific semantic classes in the modifiers. The verb can be further constrained as active or passive voice.

This representation gives CRYSTAL great flexibility in the rules that can be expressed. Identifying relevant information in a prepositional phrase might depend on the semantic class of the phrase itself and the exact word of the verb. In another case the important context may be a sequence of exact words in the subject buffer and the semantic class of the direct object. CRYSTAL has a more expressive representation than other published systems that induce text extraction rules.

PALKA [Moldovan and Kim, 1992; Moldovan *et al.*, 1993] includes an induction step similar to CRYSTAL's. PALKA constructs an initial "Frame-Phrasal pattern structure" (FP-structure) from each example clause that contains relevant information. The FP-structure constrains the root form of the verb, but allows no other exact word constraints. PALKA learns semantic constraints on each noun group (e.g. subject, direct object) in the clause, and on prepositional phrases containing relevant information. Other prepositional phrases are omitted from the FP-structure.

AutoSlog [Riloff, 1993], the precursor of CRYSTAL, also generates CN definitions from motivating instances in the text. AutoSlog uses heuristics to select certain exact words from an instance as "trigger words". Semantic constraints on the extracted buffer are set in advance by the user rather than learned. AutoSlog lacks automatic generalization to find the best level in a semantic hierarchy.

Unlike AutoSlog and PALKA, CRYSTAL makes no *a priori* decision on which constituents are to be included in its CN definitions. Any constraints including those on the verb may be retained or dropped during CRYSTAL's induction. The following section discusses the search control strategy to handle the large branching factor that results from CRYSTAL's expressive representation.

5.4 Search Control: Avoiding Intractability

Since each CN definition may have several constraints and a variety of ways to relax each constraint, there are an exponential number of generalizations possible for a given CN definition. CRYSTAL has the challenge of producing a near optimal dictionary while avoiding intractability and maintaining a rich expressiveness of its CN definitions.

CRYSTAL reduces the intractable problem of constraint relaxation to the easier problem of finding a similar CN definition. Relaxing the constraints to unify a CN definition with a similar definition has the effect of retaining the constraints shared with another valid definition and dropping accidental features of the current definition. This is also guaranteed to produce a CN definition with greater coverage than either of the definitions being unified.

Finding similar definitions efficiently is achieved by indexing the CN definitions database by verbs and by extraction buffers. In this way, CRYSTAL can retrieve a list of similar CN definitions which is small relative to the entire database. Each of these is tested with a distance metric that counts the number of relaxations required to unify the two definitions.

Testing a generalized CN definition's error rate on the training corpus is actually done on the Training Instances Database, a database of instances which have already been segmented by the BADGER sentence analyzer. This database is indexed on verbs, including the <null> verb for sentence fragments, so only a small percent of training instances are typically tested. CRYSTAL drops the constraint on exact verb only after relaxing all other constraints as far as possible, to take full advantage of the efficiency of indexing by verb.

With this control strategy, CRYSTAL is able to induce a dictionary with all CN types and subtypes from 13,000 instances in about nine minutes of clock time on a DEC ALPHA AXP 3000 using 40 MB of memory.

CRYSTAL's control structure avoids combinatorial explosion and back-tracking, but cannot guarantee that CRYSTAL makes optimal choices of which constraints to relax. This does not seem to be a problem in practice. The CN definitions that CRYSTAL generates seems to be generalized as far as possible given the training data.

6 Future Work

Our experience with a variety of domains shows that local context is generally sufficient for identifying relevant information. Sometimes, however, the context of buffers in the same clause is not sufficient. When the documents have section headers or other indications of discourse boundaries, this often influences the meaning of a phrase. A reference in the "Past Medical History" section of a hospital discharge report takes on a different meaning than a similar reference in the "Physical Examination" section.

Consider the following example, where "hypertension" is tagged as a pre-existing diagnosis. CRYSTAL's initial CN definition could be extended to include a constraint that the current section be Past Medical History. If the most similar CN definition was also from that section, this constraint would be retained, otherwise it would be dropped during induction.

PAST MEDICAL HISTORY:
Significant for <DE> hypertension </DE>

In some cases it would be useful to include context from an adjacent text segment in the extraction rule. This is particularly so, given the tendency of the BADGER sentence analyzer to break a sentence into separate segments when it encounters a conjunction. CRYSTAL's initial CN definitions could be extended to include constraints on buffers from neighboring clauses. This more distant context will usually be dropped during the induction process when a CN definition is unified with another that does not have similar constraints on a neighboring text segment.

Another improvement we intend to make to BADGER and CRYSTAL is to use a finer granularity of syntactic analysis. The current version applies CN definitions at the level of syntactic buffer (subject, verb, direct object, prepositional phrase). The CN definition does not pinpoint which simple noun phrase contains the relevant information. The next version of BADGER will segment the syntactic buffers into simple noun (or verb) phrases, also handling the scoping of negation. This will enable CRYSTAL to learn CN definitions that identify simple noun phrases and verb phrases.

Perhaps the most useful enhancement to CRYSTAL would be to add a mechanism that learns exceptions to rules. As it now stands CRYSTAL can only learn positive constraints. There are some cases where the most compact extraction rule has a constraint that a buffer *not* include a certain word or semantic class.

As it relaxes constraints on a CN definition, CRYSTAL reaches a point where there are excessive extraction errors. Rather than abandon the overly generalized CN definition, CRYSTAL could examine the negative training instances covered by the definition. A negative constraint would then be added to exclude some feature in common to several negative instances. If the CN definition is now within error tolerance, generalization would continue as before.

Suppose that training instances with the semantic class <pathological_function> in the direct object buffer are usually positive instances of "sign or symptom". However, instances with the word "chronic" in the direct object as well as <pathological_function> are tagged as "diagnosis" and are thus negative training instances for "sign or symptom". If CRYSTAL had the ability to specify a constraint that the direct object *not* contain the word "chronic", it would have a reliable extraction rule for "sign or symptom". Otherwise it would miss out on a useful generalization.

7 Conclusions

CRYSTAL operates as a fully automated knowledge acquisition tool, capable of creating a high quality dictionary of text extraction rules. Because it tests each proposed extraction rule against the training corpus, CRYSTAL is able to generalize each rule as much as possible without creating excessive extraction errors. The error tolerance parameter allows CRYSTAL to operate robustly in the face of noisy training, and allows the user to manipulate a trade-off between recall and precision.

The expressiveness of CRYSTAL's CN definitions causes an enormous branching factor in the search for an optimal level to relax the constraints in each CN definition. CRYSTAL overcomes the intractablility of the search problem by letting unification with a similar CN definition guide the induction process. This search control has worked out well in practice and allows efficient processing, while retaining a flexibility of expression that lets CRYSTAL capture subtle distinctions in its text extraction rules.

References

[Lehnert *et al.*, 1993] W. Lehnert, J. McCarthy, S. Soderland, E. Riloff, C. Cardie, J. Peterson, F. Feng, C. Dolan, and S. Goldman. University of Massachusetts/Hughes: Description of the CIRCUS system as used for MUC-5. In *Proceedings of the Fifth Message Understanding Conference (MUC-5)*, pages 277–290, 1993.

[Lehnert, 1991] W. Lehnert. Symbolic/subsymbolic sentence analysis: Exploiting the best of two worlds. In J. Barnden and J. Pollack, editors, *Advances in Connectionist and Neural Computation Theory, Vol. 1*, pages 135–164. Ablex Publishers, Norwood, NJ, 1991.

[Lindberg *et al.*, 1993] D. Lindberg, B. Humphreys, and A. McCray. Unified medical language systems. *Methods of Information in Medicine*, 32(4):281–291, 1993.

[Michalski, 1983] R. S. Michalski. A theory and methodology of inductive learning. *Artificial Intelligence*, 20:111–161, 1983.

[Mitchell, 1982] T. M. Mitchell. Generalization as search. *Artificial Intelligence*, 18:203–226, 1982.

[Moldovan and Kim, 1992] D. Moldovan and J. Kim. PALKA: A system for linguistic knowledge acquisition. Technical Report PKPL 92-8, USC Department of Electrical Engineering Systems, 1992.

[Moldovan *et al.*, 1993] D. Moldovan, S. Cha, M. Chung, T. Gallippi, K. Hendrickson, J. Kim, C. Lin, and C. Lin. USC: Description of the SNAP system used for MUC-5. In *Proceedings of the Fifth Message Understanding Conference (MUC-5)*, pages 305–319, 1993.

[Riloff, 1993] E. Riloff. Automatically constructing a dictionary for information extraction tasks. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pages 811–816, Washington, DC, July 1993. AAAI Press / MIT Press.

[Soderland *et al.*, 1995] S. Soderland, D. Fisher, J. Aseltine, and W. Lehnert. CRYSTAL: Inducing a conceptual dictionary. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, Montreal, Canada, August 1995.