AAAI 1995 Spring Symposium on Empirical Methods in Discourse Interpretation and Generation

Learning Domain-Specific Discourse Rules for Information Extraction

Stephen Soderland and **Wendy Lehnert** [*]
Department of Computer Science
University of Massachusetts
Amherst, MA 01003-4610
soderlan@cs.umass.edu lehnert@cs.umass.edu

Abstract

This paper describes a system that learns discourse rules for domain-specific analysis of unrestricted text. The goal of discourse analysis in this context is to transform locally identified references to relevant information in the text into a coherent representation of the entire text. This involves a complex series of decisions about merging coreferential objects, filtering out irrelevant information, inferring missing information, and identifying logical relations between domain objects. The Wrap-Up discourse analyzer induces a set of classifiers from a training corpus to handle these discourse decisions. Wrap-Up is fully trainable, and not only determines what classifiers are needed based on domain output specifications, but automatically selects the features needed by each classifier. Wrap-Up's classifiers blend linguistic knowledge with real world domain knowledge.

Introduction

Discourse analysis takes on a special role in a system that analyzes real-world text such as news-wire stories to identify information that is relevant to a particular information need. For a given application, domain guidelines specify what objects and attributes are considered relevant and what relationships between objects are of interest. Each domain object found in a text is represented as a case frame, with logical relationships between separate objects indicated by pointers between frames.

In this context, discourse analysis must make a number of decisions about references which a sentence analysis component has identified as relevant to the domain. Separately extracted references to the same domain object must be merged, and other references discarded as irrelevant. Pointers must be added between objects that are deemed to be logically related. Objects or attributes that are not explicitly stated in the text must in some cases be inferred.

This paper describes Wrap-Up, a trainable discourse module that uses machine learning techniques to build a set of classifiers for domain-specific discourse analysis. Wrap-Up is a fully trainable system and is unique in that it not only decides what classifiers are needed for the domain, but automatically derives the feature set for each classifier. The user supplies a definition of the objects and relationships of interest to the domain and a training corpus with hand-coded target output. Wrap-Up does the rest with no further hand coding needed to tailor the system to a new domain.

Discourse Analysis in the Microelectronics Domain

To get a sense of the processing involved in discourse analysis, consider the text fragment shown in Figure 1, which is from the microelectronics domain of the ARPA-sponsored Fifth Message Understanding Conference (MUC-5 1993). Relevant information in this domain are microchip fabrication processes, such as the x-ray lithography process mentioned in this text, as well as the companies, equipment, and devices associated with these processes.

```
IBM's Systems Integration division has
awarded Hampshire Instruments Inc. a
subcontract for x-ray mask making and
wafer exposure under the Defense Advanced
Research Projects Agency's National X-ray
Lithography Program.   ...
Under the contract, Hampshire will produce
gold-on-silicon photomasks from data
provided by IBM, print test wafers from
the masks in its Series 5000 wafer stepper
... using a laser-based soft x-ray
source  ...
Test patterns for the contract include
0.5 micron features from microprocessor
and memory devices.   ...
```

Figure 1: A sample microelectronics text

One of the main jobs of discourse analysis is to determine logical relationships between domain objects ac-

[*] This research was supported by NSF Grant no. EEC-9209623, State/ Industry/ University Cooperative Research on Intelligent Information Retrieval.

cording to domain specifications of what relationships are reportable. Domain objects are represented as case frames with pointers between objects. A microchip fabrication process can point to equipment used in the process and to devices manufactured. Equipment can point to manufacturer(s) and to equipment modules. There are four possible relationships defined between fabrication process and company: developer, manufacturer, distributor, and purchaser/user, with multiple roles possible by a company.

The relevant domain objects in the sample text include an x-ray lithography process which uses "stepper" equipment and is used to make two types of devices, microprocessors and memory chips. The stepper in turn has "radiation-source" equipment as a submodule. Although two companies and a government agency were mentioned, only Hampshire Instruments Inc. plays a direct role in the lithography process, so IBM and DARPA are not considered relevant and should be ignored.

The desired output for this text is shown in Figure 2. A "microelectronics-capability" object is created to show the relationship between company and processes, with Hampshire Instruments as the developer of the lithography process. Lithography points to the stepper, which points to the radiation source equipment and back to Hampshire Instruments. The lithography also points to two devices.

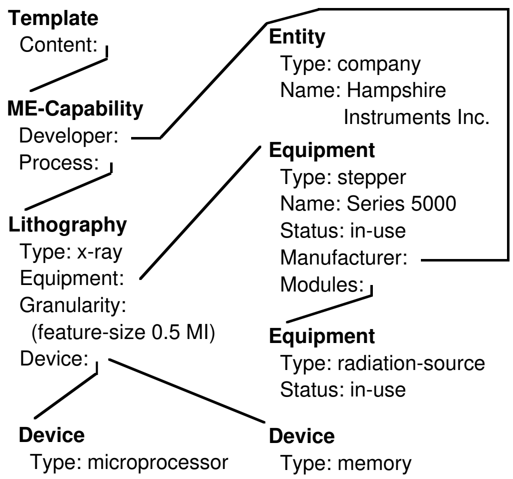

Figure 2: Output of analysis of the sample text

The following section describes how Wrap-Up uses decision trees to tailor its discourse processing to each new domain, how features are generated for these trees, and how the trees are trained from hand-coded target output.

The Wrap-Up Discourse Analyzer

Wrap-Up is a domain-independent framework that is instantiated with a set of classifiers to handle domain-specific discourse analysis. During its training phase, Wrap-Up derives a set of ID3 decision trees (Quinlan 1986) from a representative set of training texts. Each training text has a hand-coded output key indicating the objects and relationships that should be identified for that text.

The input to Wrap-Up is a set of domain objects identified by the University of Massachusetts CIRCUS sentence analyzer (Lehnert 1990; Lehnert et al. 1993). CIRCUS uses domain-specific extraction patterns to create a "concept node" (CN) for each noun phrase that contains relevant information (Riloff 1993). Each CN has a case frame for the extracted information along with the position of the reference and a list of extraction patterns used.

Wrap-Up must merge information from coreferential CN's, discard information that was erroneously identified during sentence analysis, and determine logical relationships between objects. A key to using machine learning classifiers for a complex task such as discourse analysis is to break the processing down into a number of small decisions, each handled by a separate classifier.

The Wrap-Up algorithm consists of six steps, with a set of classifiers built to guide processing in each step.

1. Filtering out spuriously extracted information
2. Merging coreferential object attributes
3. Linking logically related objects
4. Splitting objects with multiple pointers
5. Inferring missing objects
6. Adding default slot values

Wrap-Up automatically determines the trees needed for each step from a domain output specification, which consists of a list of the objects and possible pointers between objects. The filtering stage has a separate classifier for each slot of each domain object; the merging stage has a separate classifier for each type of object; the linking stage has a separate classifier for every possible pointer relationship in the output specifications, and so forth. A total of 91 decision trees were used for the microelectronics domain.

The following section illustrates how decision trees guide Wrap-Up's processing.

Decision Trees for Discourse Analysis

During analysis of a text, Wrap-Up creates an instance for one of its classifiers each time it encounters a discourse decision in the text. For example, CIRCUS identified an x-ray lithography object in the Figure 1 text as well as two equipment objects, one of type "stepper" and the other of type "radiation-source". One of the decisions in the linking stage of Wrap-Up is whether to add a pointer from the lithography process to either or both equipment objects.

Wrap-Up creates a separate instance for each pair of possibly linked objects, one for the lithography and the radiation source pair, and another for lithography and the stepper equipment. Each instance is passed to a "Lithography-Equipment-Link" decision tree and a pointer is added if the tree returns "positive".

An instance has features for each of the pair of objects, such as the features "obj1:type-x-ray" and "obj2:type-radiation-source" which express information from slots of the CN case frames. Other features are derived from the CN definitions (a combination of keywords and linguistic patterns) that CIRCUS used to identify references to the object. These features include "obj1:keyword-x-ray", "obj1:subcontract-for-X", and "obj1:using-X". The instance also includes features indicating the relative position of references to the two objects, such as "same-sentence" or "one-common-phrase". More details about feature generation are provided in the following section.

Figure 3 shows a portion of the Lithography-Equipment-Links tree. ID3 recursively selects features to partition the training instances according to an information gain metric, choosing "same-sentence" as the root of the tree. The baseline probability of a link from lithography to equipment was 31%, but only 9% of the training instances were positive when references to both objects did not occur in the same sentence.

ID3 selects the feature "obj2:type-radiation-source" to partition the instances where "same-sentence" is true. All training instances with equipment type "radiation-source" are negative, since a radiation source can only serve as a sub-module of other equipment and not be directly linked to a lithography process. These two tree nodes are sufficient to classify the lithography-radiation-source instance as negative.

Further tests are needed for equipment other than radiation source. If the lithography process was identified using the keyword "stepper" the probability of a link increases to 77%. This is not the case here, and the next true-valued feature is "obj2:modules", since a previous linking decision has added a pointer to the radiation source from the "modules" slot of the stepper's case frame. After branching on true for lithography type "x-ray" the lithography-stepper instance is classified as positive.

These paths through the tree illustrate useful domain knowledge learned by Wrap-Up. Do not add a pointer from lithography to equipment of type radiation source, even if both references are cosentential. A pointer should be added when both references are from the same sentence, the equipment has a module, and the lithography is of type "x-ray". This tree illustrates how Wrap-Up fluidly combines real world knowledge about what equipment can be used for lithography with rules about the relative position of references and about specific lexical context.

Figure 4 shows a "Lithography-Merging" decision tree from stage 2 of Wrap-Up. In virtually all the positive training instances object 1 has the attribute type, such as "x-ray" lithography, and object 2 has the attribute granularity, such as "feature-size 0.5 MI". Multiple references to objects of the same type have already been merged without consulting a classifier, such as all the references to "x-ray" lithography. Pairs of lithography objects of different type are not mergeable.

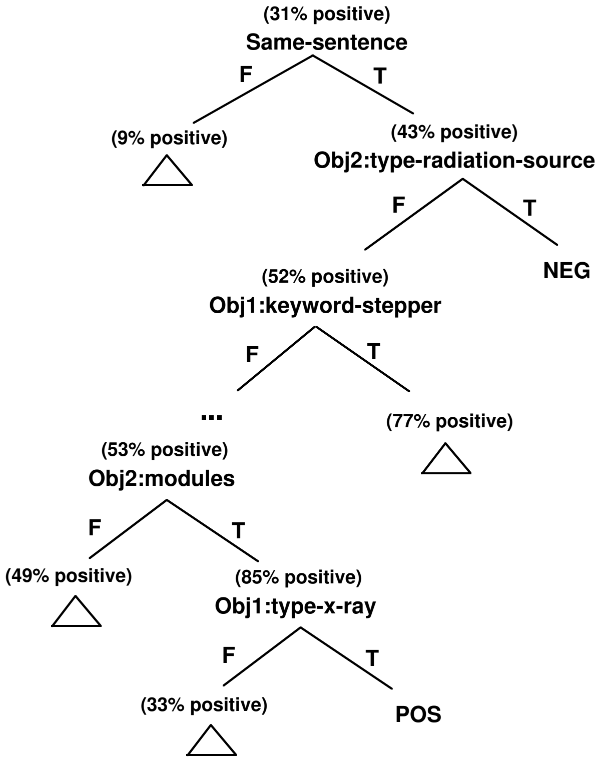

Figure 3: Decision tree for adding pointers from lithography (object 1) to equipment (object 2)

ID3 has selected "obj2:granularity" as the root of the tree and has a variety of tests at other nodes, including linguistic patterns such as "resolutions of X" and "using X". The domain knowledge encapsulated in the this "Lithography-Merge" decision tree indicates that positive instances of merging lithography with granularity often occur for "I-line" and "UV" lithography and are often associated with the keyword "features".

The two decision trees shown in this section are from only two of Wrap-Up's six stages of discourse processing. The filtering stage has a classifier to judge the validity of each slot of each CN, possibly discarding slots as irrelevant if a tree returns "negative". After the merging and linking stages, Wrap-Up has a stage that considers objects with multiple pointers. In some cases the object is split into multiple copies, each with a single pointer.

The next stage considers "orphaned" objects, those with no other objects pointing to them, and may infer an object to point to an orphan. Trees for this stage return a classification that specifies what type of object to infer. The last stage adds context-sensitive default values to empty slots, such as the status of "in-use" or "in-development" for equipment objects.

For all of these classifiers it is critical that the feature encoding is expressive enough to support the necessary discriminations. We now look at Wrap-Up's mechanism for generating features automatically.

Automatic Feature Generation

When Wrap-Up creates an instance for an object or a pair of objects, it encodes as much of the information passed to it by the sentence analyzer as possible. Each "concept node" extracted by CIRCUS has a case frame with a slot for each attribute, such as equipment-type or equipment-name. CIRCUS also passes along the position of each reference in the text along with the CN definitions used to identify that reference, from a domain-specific dictionary of extraction patterns such as "using-X", "subcontract-for-X", or "keyword-stepper".

Instances are encoded as a list of binary features, some features indicating the slot-values of each object, some expressing the relative position in the text of the nearest references to the two objects, some showing the CN-definition patterns for each reference, and the number of such patterns.

Here is an instance from the sample text for the Lithography-Equipment-Links tree.

```
(obj1:type-x-ray . t) (obj1:type . t)
(obj1:granularity-feature-size . t)
(obj1:granularity-MI . t) (obj1:granularity . t)
(obj2:type-stepper . t) (obj2:type . t)
(obj2:name-series . t) (obj2:name-5000 . t)
(obj2:name .t) (obj2:modules . t)
(obj2:manufacturer . t) (obj1:keyword-x-ray . t)
(obj1:using-X . t) (obj1:subcontract-for-X . t)
(obj1:cn-count>=2 . t) (obj2:masks-in-X . t)
(obj2:keyword-stepper . t) (obj2:awarded-X . t)
(obj2:keyword-inc . t) (obj2:cn-count>=2 . t)
(same-sentence . t) (common-noun-phrase . t)
```

Object-1 is a lithography object with type x-ray and object-2 is equipment with type "stepper". By the time Wrap-Up creates this instance, the granularity "feature-size 0.5 MI" has already been merged with the x-ray lithography object and pointers have been added from the stepper equipment to Hampshire Instruments as manufacturer and to the radiation source as module.

CN definition features include the patterns "using X", "subcontract for X", and keyword "x-ray" that identified x-ray lithography. Stepper equipment was extracted with the pattern "masks in X" and the keyword "stepper". The equipment object also inherits CN definition features from its manufacturer pointer to Hampshire Instruments: "awarded X" and keyword "Inc".

Some features that encode specific linguistic patterns or equipment names may have low frequency and contribute only noise to the classification. Wrap-Up prunes the features by setting a minimum number of texts that a feature must be found in, and discards features below this threshold.

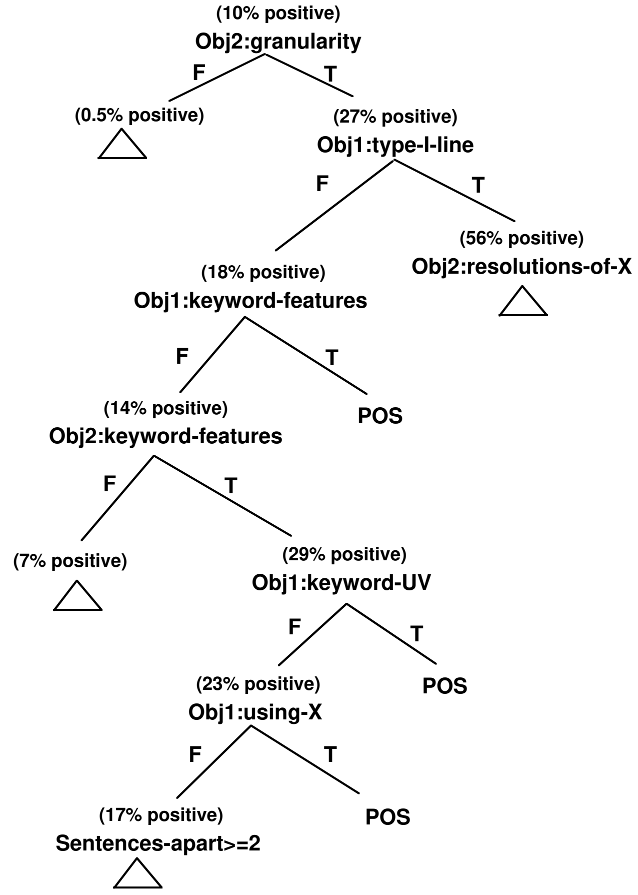

Figure 4: Decision tree to merge lithography attributes

Having looked briefly at Wrap-Up's mechanism for generating features, we consider how Wrap-Up uses a corpus of training texts to create its training instances.

Creating Training Instances

ID3 is a supervised learning algorithm and requires a set of training instances, each labeled with the correct classification for that instance. The user supplies Wrap-Up with a representative set of training texts and hand-coded target output for each text. Wrap-Up begins its tree-building phase by passing each training text to the sentence analyzer, which creates a set of objects representing the extracted information. The output from sentence analysis forms the initial input to Wrap-Up's first stage. Wrap-Up encodes instances and builds trees for this stage, then uses trees from stage one to build trees for stage two, and so forth until trees have been built for all six stages.

As it encodes instances, Wrap-Up repeatedly consults the target output to assign a classification for

Classifier	Ftrs	Insts	% Pos	Accuracy	Precision	Recall
Filtering:						
device type	214	1029	40.3	73.2	61.5	90.1
entity name	359	3282	34.8	75.6	63.1	71.7
equipment type	181	1284	62.6	86.6	82.4	99.9
etching type	40	295	32.9	60.0	44.4	86.6
layering type	78	561	62.2	70.9	70.9	90.3
lithography type	96	538	57.4	74.7	71.4	93.5
packaging type	127	637	55.2	69.5	67.6	86.4
Merging:						
device attributes	736	1433	17.9	86.9	67.2	52.5
entity attributes	899	7278	39.6	62.4	51.9	70.2
equipment attributes	426	1229	28.1	86.5	77.5	73.7
lithography attributes	234	361	11.1	88.6	48.6	42.5
Linking:						
equipment manufacturer	623	1966	22.5	81.1	56.4	69.5
equipment modules	554	990	6.9	94.0	58.3	50.7
lithography device	349	563	24.2	83.9	68.9	60.3
lithography equipment	489	774	29.9	78.8	63.6	68.0
lithography developer	696	1312	9.2	86.4	22.9	19.8
lithography distributor	696	1312	8.0	91.5	46.2	34.3
lithography manufacturer	696	1312	9.7	90.9	53.9	43.3
lithography purchaser/user	696	1312	8.8	86.6	28.9	35.3

Figure 5: Performance of individual decision trees

each training instance. For the merging stage an instance is created for each pair of objects of the same type. If both objects can be mapped to the same object in the target output, the instance is classified as positive. The links stage will similarly encode an instance for object A and object B, then look in the target output for an object matching A that has a pointer to an object matching B.

There are limits to the supervision provided by target output of the granularity used by Wrap-Up. In particular it is inadequate for learning coreference merging and anaphora resolution. Suppose a lengthy text has several references to lithography processes, as well as generic references to new “technology”, a “process”, and various pronominal references. Some of these references may indeed equate to a lithography object in the target output, but others may be vague references that are not reportable by domain guidelines. If the target output has more than one process, it is not clear how to map them to specific references in the text.

Wrap-Up faces a difficult problem dealing with spurious objects that were extracted by the sentence analyzer, but were not found in the target output. Some of these were legitimate references to a company, such as IBM in our example text, that are not linked directly to a microchip fabrication process and should be discarded. Devices and equipment should likewise be discarded if it is not associated with a specific process and processes discarded that are not linked to a company. About half of the objects identified by CIRCUS in this domain were spurious. Those that persist past Wrap-Up's filtering step are a source of noise in the training for later steps.

System Performance

As previously reported (Soderland and Lehnert 1994a, 1994b) the full system with Wrap-Up compares favorably with the discourse module used in the MUC-5 evaluation by UMass, which was only partially trainable. Here we will look at the performance of Wrap-Up from the point-of-view of individual classifiers, as shown in Figure 5.

Wrap-Up's performance is best evaluated in terms of the metrics “recall” and “precision”. Recall is the percentage of possible information that was reported. Precision is the percent correct of the reported information. If there are 400 positive instances and the classifier identifies 300 of them as positive, recall is 75%. If the classifier had 300 false positives as well, its precision is 50%.

The high degree of spurious input to Wrap-Up can be seen from the low percent of positive instances for the filtering stage classifiers. Only 40% of the device types and 35% of the company names identified by CIRCUS were relevant to the domain. Wrap-Up's filtering stage is able to raise precision above that of CIRCUS output by discarding more than half of the spurious objects, although at the cost of discarding some valid objects as well.

Classifiers for later stages of Wrap-Up were hampered by noisy training due to the spurious objects that were retained and more seriously by the coarse granularity of sentence analysis. Information that is not reportable according to domain guidelines (pronominal and generic references), is often critically important to inferring relationships between domain objects. Wrap-Up encoded in its instances as much information about local linguistic context as was provided by the sentence analysis component, but this was insufficient for deep reasoning.

Conclusions

The goal of Wrap-Up is to be a fully trainable discourse component, a turnkey system that can be tailored to new information needs by users who have no special linguistic or technical expertise. The user defines an information need and output structure, then provides a training corpus of representative texts with hand coded target output for each text.

Wrap-Up takes it from there and instantiates a fully functional discourse analyzer for the new domain with no further customization needed by the user. Wrap-Up is the first fully trainable system to handle discourse processing, automatically determining the classifiers needed for the domain output structure and deriving the feature set for each classifier from sentence analyzer output.

While creating a sufficient training corpus represents a labor-intensive investment on the part of domain experts, it is easier to generate a few hundred answer keys than it is to write down explicit and comprehensive domain guidelines. Once available, this corpus of training texts can be used repeatedly for knowledge acquisition at all levels of processing. The same training corpus was used to induce a dictionary of CN definitions used by CIRCUS in sentence analysis (Riloff 1993).

A thousand texts provided the training for Wrap-Up in the microelectronics domain. The number of actual training instances provided by these texts varied for different classifiers, with some reaching saturation from as few as two hundred texts and others that dealt with less frequent domain objects and relationships, still under trained at a thousand texts.

Wrap-Up differs from other work on discourse, which has often involved tracking shifts in topic and in the speaker/writer's goals (Grosz and Sidner 1986; Liddy et al. 1993) or in resolving anaphoric references (Hobbs 1978). Domain-specific discourse analysis that processes unrestricted text may concern itself with some of these issues, but only as a means to its main objective of transforming bits and pieces of extracted information into a coherent representation.

Wrap-Up does not have available an in-depth analysis of the text, or a model of the writer's goals and beliefs. Wrap-Up relies instead on the information supplied by sentence analysis, which gives the local context of domain objects found in the text. A domain-specific text analysis system concerns itself only with clauses that contain information relevant to the information extraction task and ignores other portions of text.

Wrap-Up must make discourse decisions robustly in the face of unrestricted text and incomplete knowledge. Much of its discourse processing is dominated by real-world knowledge about the domain objects involved. Other processing is based on the relative position in the text of two references or on the keyword or phrase used in that reference. Wrap-Up's classifiers blend together these types of knowledge based on actual usage in a representative text corpus.

References

Grosz, B.; Sidner C. 1986. Attention, intention and the structure of discourse, 175-204. *Computational Linguistics, 12(3).*

Hobbs, J. 1978. Resolving Pronoun References, 311-338. *Lingua, 44(4).*

Lehnert, W. 1990. Symbolic/Subsymbolic Sentence Analysis: Exploiting the Best of Two Worlds, 151-158. *Advances in Connectionist and Neural Computation Theory. vol. 1.*. Norwood, NJ: Ablex Publishing.

Lehnert, W.; McCarthy, J.; Soderland, S.; Riloff, E.; Cardie, C.; Peterson, J.; Feng, F; Dolan, C.; Goldman, S. 1993. UMass/Hughes: Description of the CIRCUS System as Used for MUC-5, 257-259. In Proceedings of the Fifth Message Understanding Conference. Morgan Kaufmann Publishers.

Liddy, L.; McVearry, K.; Paik, W.; Yu, E.; McKenna, M. 1993. Development, Implementation, and Testing of a Discourse Model for Newspaper Texts, 159-164. In Proceedings of the Human Language Technology Workshop. Morgan Kaufmann Publishers.

MUC-5. 1993. Proceedings of the Fifth Message Understanding Conference. Morgan Kaufmann Publishers.

Quinlan, J.R. 1986. Induction of Decision Trees, 81-106. *Machine Learning, 1.*

Riloff, E. 1993. Automatically Constructing a Dictionary for Information Extraction Tasks, 811-816. In Proceedings of the Eleventh National Conference on Artificial Intelligence.

Soderland, S. and Lehnert, W. 1994a. Corpus-Driven Knowledge Acquisition for Discourse Analysis, 827-832. In Proceedings of the Twelfth National Conference on Artificial Intelligence.

Soderland, S. and Lehnert, W. 1994b. Wrap-Up: a Trainable Discourse Module for Information Extraction, 131-158. *Journal of Artificial Intelligence Research, 2.*