Proceedings of the Conference on Empirical Methods in Natural Language Processing, 113–126, University of Pennsylvania, 1996.

Automating Feature Set Selection for Case-Based Learning of Linguistic Knowledge

Claire Cardie
Department of Computer Science
Cornell University
Ithaca, NY 14853–7501
E-mail: cardie@cs.cornell.edu

Abstract

This paper addresses the issue of "algorithm vs. representation" for case-based learning of linguistic knowledge. We first present empirical evidence that the success of case-based learning methods for natural language processing tasks depends to a large degree on the feature set used to describe the training instances. Next, we present a technique for automating feature set selection for case-based learning of linguistic knowledge. Given as input a baseline case representation, the method modifies the representation in response to a number of predefined linguistic biases by adding, deleting, and weighting features appropriately. We apply the linguistic bias approach to feature set selection to the problem of relative pronoun disambiguation and show that the case-based learning algorithm improves as relevant biases are incorporated into the underlying instance representation. Finally, we argue that the linguistic bias approach to feature set selection offers new possibilities for case-based learning of natural language: it simplifies the process of instance representation design and, in theory, obviates the need for separate instance representations for each linguistic knowledge acquisition task. More importantly, the approach offers a mechanism for explicitly combining the frequency information available from corpus-based techniques with linguistic bias information employed in traditional linguistic and knowledge-based approaches to natural language processing.

Introduction

Standard symbolic machine learning techniques have been successfully applied to a number of tasks in natural language processing (NLP). Examples include the use of decision trees for syntactic analysis (Magerman, 1995), coreference (Aone and Bennett, 1995; McCarthy and Lehnert, 1995), and cue phrase identification (Litman, 1994); the use of inductive logic programming for learning semantic grammars and building prolog parsers (Zelle and Mooney, 1994; Zelle and Mooney, 1993); the use of conceptual clustering algorithms for relative pronoun resolution (Cardie, 1992a; Cardie 1992b), and the use of case-based learning techniques for lexical tagging tasks (Cardie, 1993a; Daelemans et al., submitted). In theory, both statistical and machine learning techniques can significantly reduce the knowledge-engineering effort for building large-scale NLP systems: they offer an automatic means for acquiring robust heuristics for a host of lexical and structural disambiguation tasks. It is well-known in the machine learning community, however, that the success of a learning algorithm depends critically on the representation used to describe the training and test instances (Almuallim and Dietterich, 1991, Langley and Sage, in press). Unfortunately, the task of designing an appropriate instance representation — also known as *feature set selection* — can be extraordinarily difficult, time-consuming, and knowledge-intensive (Quinlan, 1983). This poses a problem for current statistical and machine learning approaches to natural language understanding where a new instance representation is typically required for each linguistic task tackled.

This paper addresses the role of the underlying instance representation for one class of symbolic machine learning algorithm as applied to natural language understanding tasks, that of *case-based learning* (CBL). In general, case-based learning algorithms (e.g., instance-based learning (Aha et al., 1991), case-based reasoning (Riesbeck and Schank, 1989, Kolodner, 1993), memory-based reasoning (Stanfill and Waltz, 1986) solve problems by first creating a case base of previous problem-solving episodes. Then, when a new problem is encountered, the "most similar" case is retrieved from the case base and used to solve the novel problem. The retrieved case can either be used directly or after one or more modifications to adapt it to the current problem-solving situation. Case-based learning algorithms have been used in NLP for context-sensitive parsing (Simmons and Yu, 1992), for text categorization (Riloff and Lehnert, 1994); for lexical tagging tasks like part-of-speech tagging and semantic feature tagging (Daelemans et al., submitted, Cardie, 1994, Cardie, 1993a); for semantic interpretation (e.g., concept extraction (Cardie, 1994, Cardie, 1993a)); and for a number of low-level language acquisition tasks, including stress acquisition (Daelemans et al., 1994) and grapheme-to-phoneme conversion (Bosch and Daelemans, 1993). In the sections below, we first present empirical evidence that the success of case-based learning methods for natural language processing tasks depends to a large degree on the feature set used to describe the training instances. Next, we present a technique for automating feature set selection for case-based learning of linguistic knowledge. Given as input a baseline instance representation comprised of both relevant and irrelevant attributes, the method modifies the representation in response to any of a number of predefined linguistic biases. More specifically, the technique uses linguistic biases to discard irrelevant features from the representation, to add new features to the representation, and to weight features appropriately. We then apply the linguistic bias approach to feature set selection in one natural language learning task — the relative pronoun (RP) disambiguation task from Cardie (1992a, 1992b) . Experiments indicate that the case-based learning algorithm improves on the relative pronoun task as relevant biases are incorporated into the underlying instance representation. Furthermore, using the modified instance representation, the case-based learning algorithm is able to outperform a set of hand-coded heuristics designed for the same task.

Finally, we argue that the linguistic bias approach to feature set selection offers new possibilities for case-based learning of natural language:

- It provides a natural mechanism for combining the frequency information available from corpus-based NLP techniques with linguistic bias information employed in traditional linguistic and knowledge-based approaches to language processing. The development of computational models of language processing that combine frequencies and linguistic biases has been noted by Pereira (Pereira, 1994) as an important area of research in corpus-based NLP.

- The linguistic bias approach to feature set selection simplifies and shortens the process of designing an appropriate instance representation for individual natural language learning tasks. System developers can safely include features for *all* available knowledge sources in the baseline instance representation — the irrelevant ones will be discarded automatically.

- By adopting the automated approach to feature set selection for CBL of linguistic knowledge, the same underlying instance representation can, in theory, be used across many linguistic knowledge acquisition tasks. A separate instance representation need not be designed each time we want to apply the learning algorithm to a new problem in natural language understanding.

The remainder of the paper is organized as follows. The section below describes the basic case-based learning algorithm used throughout the paper. The following section determines the role of the underlying instance representation in case-based learning of natural language by comparing the accuracy of the CBL algorithm on a number of natural language learning tasks using different instance representations. Next, we present the linguistic bias approach to feature set selection and applies the technique to the relative pronoun disambiguation task. We conclude with a discussion of the general implications of the linguistic bias approach to feature set selection for case-based learning of natural language.

The Basic Case-Based Learning Algorithm

Throughout the paper, we employ a simple k-nearest neighbor case-based learning algorithm. In addition, we assume that the learning algorithm is embedded in a parser or larger NLP system and, hence, has access to all knowledge sources that are available to the NLP system.

In case-based approaches to natural language understanding, the goal of the training phase is to collect a set of cases that describe ambiguity resolution episodes for a particular problem in text analysis. To do this, a small set of sentences is first selected randomly from an annotated training corpus. Next, the sentence analyzer processes the training sentences and creates a training case every time an instance of the ambiguity occurs. To learn heuristics for prepositional phrase attachment, for example, the parser would create a case whenever it recognizes a prepositional phrase. Each case is a set of features, or attribute-value pairs, that encode the context in which the ambiguity was encountered. In general, the *context features* represent the state of the parser at the point of the ambiguity. In addition, each case is annotated with one or more pieces of "class" information that describe how the ambiguity was resolved in the current example. We will refer to these as *solution features*. For lexical tagging tasks, for example, the class information is the syntactic or semantic category associated with the current word; for structural attachment decisions, the class information indicates the position of the preferred attachment point. As cases are created, they are stored in a case base.

After training, the system can use the case base to resolve ambiguities in novel sentences. Whenever the sentence analyzer encounters an ambiguity, it creates a problem case, automatically filling in its context portion based on the state of the natural language system at the point of the ambiguity. The structure of a problem case is identical to that of a training case except that the solution part of the case is missing. Next, the case retrieval algorithm compares the problem case to those stored in the case base, finds the most similar training case, and then uses the class information to resolve the current ambiguity.

The experiments described below employ the following case retrieval algorithm:

1. Compare the problem case, X, to each case, Y, in the case base and calculate for each pair:

$$\sum_{i=1}^{|N|} match(X_{N_i}, Y_{N_i})$$

   where N is the set of features used to describe all instances, N_i is the ith feature in the ordered set, X_{N_i} is the value of N_i in the problem case, Y_{N_i} is the value of N_i in the training case, and $match(a, b)$ is a function that returns 1 if a and b are equal and 0 otherwise.

2. Return the k highest-scoring cases plus any ties.
3. Let the retrieved cases vote on the predicted class (solution) value and use that value to resolve the ambiguity for X. We use a simple majority vote and break ties randomly.

The case retrieval algorithm is essentially a simple k-nearest neighbors algorithm, with minor modifications to handle symbolic features.

The Role of Representation in Case-Based Learning of Linguistic Knowledge

This section explores the role of the instance representation in case-based learning of natural language. In particular, it should be clear that the basic case-based learning algorithm will perform poorly when cases contain many irrelevant attributes (Aha et al., 1991, Aha, 1989). Unfortunately, deciding which features are important for a particular learning task is difficult, especially when interactions among potentially relevant features are unpredictable.

In previous work (Cardie, 1994), for example, we applied the above case-based learning algorithm to a number of problems in sentence analysis both with and without mechanisms for feature set selection. Table 1 summarizes our results for simultaneous part-of-speech and semantic class (i.e., word sense) tagging.[1] Details regarding the experiments are included as part of Table 1. It shows that tagging accuracy increases significantly when access to the available feature set is appropriately limited. More specifically, each tagging decision is initially described in the case representation in terms of 33 features: 22 local context features encode syntactic and semantic information for the words within a five-word window centered on the current word; 11 global context features encode information for any major syntactic constituents that have been recognized (e.g., semantic class and concept activation information for the subject, direct object, verb). The general idea behind the representation of context is to include any information available to the parser that might be useful for inferring the part of speech and semantic features of the current word. Results for the CBL algorithm using all 33 features are shown in the column labeled "w/o feature selection." Intuitively, however, it seems that very different subsets of the feature set may be useful for part-of-speech prediction and semantic class prediction. Not surprisingly, the accuracy of the CBL algorithm increases when task-specific subsets of the original feature set are used instead of all of the available features (see the last column of Table 1).

The task-specific subsets for the lexical tagging experiments of Table 1 were obtained automatically using the C4.5 decision tree algorithm (Quinlan, 1992) as described in Cardie(1993b). Very briefly, in addition to storing training cases in the case base, we use them to train a decision tree for each of the selected lexical tasks. Features that appear in the pruned decision tree are assumed to be relevant to the task; features that are missing from the tree are assumed to be unnecessary for the task. The feature sets proposed by C4.5 reduce the number of attributes used in the case retrieval algorithm from 33 to an average of 14, 11, and 15 features for part-of-speech, general semantic class, and specific semantic class tagging, respectively.[2] In addition, this automated approach to feature selection outperforms feature sets chosen by hand (Cardie, 1993b): the automated approach locates features that human experts consider mildly relevant to the task at best, but that, in practice, provide statistically reliable cues for the prediction

[1] Word senses were represented in terms of a two-level domain-specific semantic feature hierarchy.

[2] A more sophisticated variation of this approach has been used by Daelemans et al. (1993) to provide weights on features rather than to eliminate features. It is able to improve on our semantic feature tagging results by a few percentage points.

Table 1: **Results for Lexical Tagging Using Case-Based Learning With and Without Feature Set Selection.** (All experiments draw training and test cases from a base set of 120 sentences from the MUC/TIPSTER Joint Ventures corpus (MUC-5, 1994). A relatively small corpus was used because the domain-specific semantic class tags and the tags for another lexical tagging task (not described here) were not available as part of any existing annotated corpus and had to be provided manually. The results presented are 10-fold cross validation averages using the same breakdown of training/test set cases for each experiment. The parser used to generate training and test cases was the CIRCUS system (Cardie and Lehnert, 1991; Lehnert, 1990). The case retrieval algorithm was modified slightly to prefer cases among the top $k = 10$ cases that match the current word. A more detailed description of the experiments and an analysis of the results can be found in Cardie(1993a, 1994).)

Lexical Tagging Task	Number of Classes	Examples of Class Information	CBL Algorithm	
			w/o feature selection (% correct)	w/feature selection (% correct)
part-of-speech	18	noun, gerund, noun modifier, adverb	91.1	95.0
general semantic class	14	joint venture entity, human, facility	67.1	80.6
specific semantic class	42	company name, government, factory	73.7	85.5

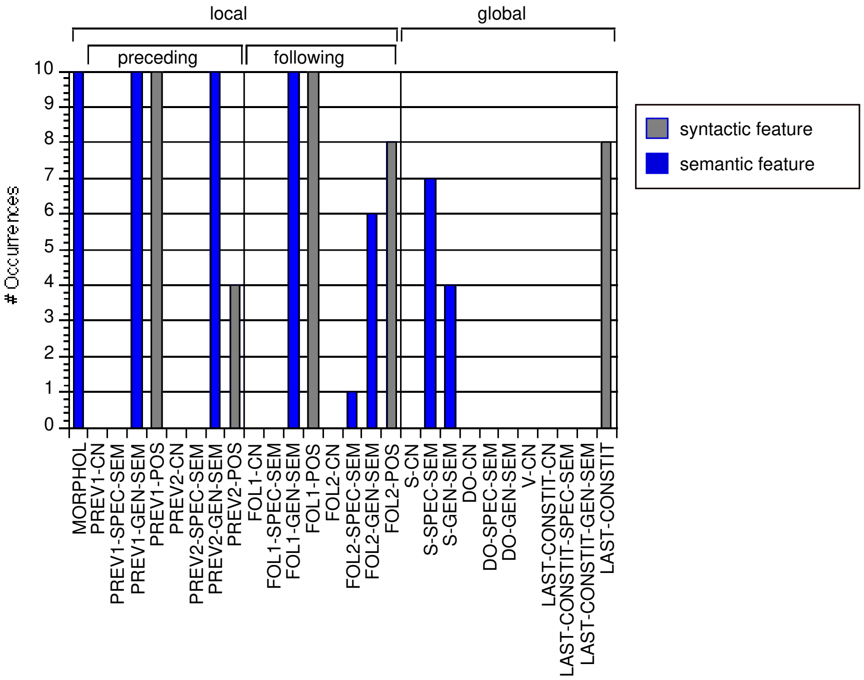

Figure 1: **Histogram of Relevant Context Features for Part-of-Speech Tagging.** (In the graph, *prev* and *fol* refer to the preceding and following lexical items; *gen-sem* and *spec-sem* refer to general and specific semantic class values; *cn* refers to concept/case-frame activation; *morphol* refers to the morphology of the word to be tagged; *s*, *do*, *v*, and *last-constit* refer to the subject, direct object, verb, and last low-level constituent (i.e., noun phrase, verb, prepositional phrase), respectively.)

task. Among the features deemed most important for part-of-speech tagging, for example, included the general semantic class of the two preceding words, the general semantic class of the following word, and the semantic class of the subject of the current clause. This is in addition to more obviously relevant features: e.g., the morphology of the current word, the part of speech of the preceding and following words. A histogram of the relevant features for part-of-speech tagging across the 10 folds of the cross-validation experiments are shown in Figure 1.

Based on the experiments described in this section, we can conclude that the overall accuracy of case-based learning of linguistic knowledge depends to a large degree on the feature set used in the case representation. Moreover, automatic approaches to feature set selection can outperform feature sets chosen manually by taking advantage of statistical relationships in the data that are difficult for humans to predict and that may be idiosyncrasies of the task and data set at hand.

Using Linguistic and Cognitive Biases for Feature Set Selection

We saw in the last section that the performance case-based learning algorithms degrades when features irrelevant to the learning task are included in the underlying instance representation. As a result, the basic CBL algorithm for lexical tagging tasks was augmented with a decision tree algorithm whose job it was to discard irrelevant features from the case representation. This section presents a new technique for feature set selection for case-based learning of natural language. The new approach is potentially more powerful than the decision tree method in that it can improve a baseline case representation in three ways rather than one:

1. It discards irrelevant features from the representation.
2. It determines the relative importance of relevant features.
3. It has a limited capability for adding new features when the existing ones are inadequate for the learning task.

Furthermore, the algorithm relies on an inductive bias that may be more appropriate to problems in natural language understanding than the information gain metric used in the C4.5 decision tree system: our linguistic bias approach to feature set selection automatically and explicitly encodes any of a predefined set of linguistic biases and cognitive processing limitations into a baseline instance representation.

Thus far, we have incorporated three such biases into the feature set selection algorithm: (1) a recency bias, (2) a restricted memory bias, and (3) a subject accessibility bias. Modifications to the instance representation in response to these biases either *directly* or *indirectly* change the feature set used to describe all instances. Direct changes to the representation are made by adding or deleting features; indirect changes modify a weight associated with each feature.

In the paragraphs below, we describe these biases and show how they can be used to modify the case representation for the task of relative pronoun (RP) disambiguation. The goal of the learning algorithm for relative pronoun disambiguation is: (1) to determine whether the wh-word is being used as a relative pronoun, and, if it is, (2) to determine which constituents comprise the antecedent. In the sentence,

I saw the boy *who* won the contest.

for example, the CBL system must decide that "who" is a relative pronoun that refers to "the boy."

The baseline instance representation for the relative pronoun task is similar to the one used for the lexical tagging tasks. The main difference is that additional global context features are included in the case representation — namely, the parser includes one attribute-value pair for every constituent in the clause that precedes the relative pronoun. Figure 2 shows a portion of three relative pronoun disambiguation cases using the baseline case representation. Each constituent is described in terms of its syntactic class and its position in the sentence as it was encountered by the CIRCUS parser. The value for each feature provides the phrase's semantic class. The class information assigned to each case describes the location of the correct antecedent. Note that no attachment decisions have been made by the parser; these will be made by learning algorithm as needed. In our current implementation, the learning algorithm, rather than the parser, is also responsible for interpreting any conjunctions and appositives that are part of the antecedent as shown in sentences S2 and S3 of Figure 2.

The case representation for the RP task creates a minor problem for the CBL algorithm: no two instances are guaranteed to have the same features. Sentences that exhibit a direct object, for example, will have a "direct object" feature; sentences that have no direct object will contain no "direct object" feature. As a result, we require that all instances are described in terms of a normalized set of features. To do this, the algorithm keeps track of every attribute that occurs in the training instances and augments the

S1:		[The man] [from Oklahoma] [,] who ...
	features:	*(s human) (s-pp1 location) (prev1-syntactic-type comma) ...*
	class:	*s*
	(The antecedent is the subject.)	
S2:		[I] [thank] [Nike] [and] [Reebok] [,] who ...
	features:	*(s human) (v exists) (do name) (do-np1 name) (prev1-syntactic-type comma) ...*
	class:	*do + do-np1*
	(The antecedent involves two constituents.)	
S3:		[I] [thank] [our sponsor] [,] [GE] [,] who ...
	features:	*(s human) (v exists) (do entity) (do-np1 name) (prev1-syntactic-type comma) ...*
	class:	*do-np1* $\vee$ *do*
	(There are two semantically legal antecedents.)	

Figure 2: **Baseline Instance Representation.**

training and test instances to include every feature of the normalized feature set, filling in a *nil* value if the feature does not apply for the particular instance. Unfortunately, this means that most of the features in a normalized case will be one of these "missing features." To ensure that the case retrieval algorithm focuses on features that are present rather than missing from the problem case, we also modify the original case retrieval algorithm to award full credit for matches on features present in the problem case and to allow partial credit for matches on missing features. This is accomplished by associating with each feature a weight that indicates the importance of the feature in determining case similarity and by using a *weighted* nearest-neighbor case retrieval algorithm:

1. Set the weight, w_f, associated with each feature, f, in the normalized feature set[3]:

   w_f = 0.2 if f is missing from the (unnormalized) problem case,
   w_f = 1 otherwise.

2. Compare the problem case, P, to each training case, T, in the case base and calculate, for each pair:

   $$\sum_{i=1}^{|N|} w_{N_i} * match(P_{N_i}, T_{N_i})$$

   where N is the normalized feature set, N_i is the ith feature in N, P_{N_i} is the value of N_i in the problem case, T_{N_i} is the value of N_i in the training case, and $match(a,b)$ is a function that returns 1 if a and b are equal and 0 otherwise.

3. Return the case with the highest score as well as all ties.

4. Let the retrieved cases vote on the value of the antecedent. Again, we use a simple majority vote and break ties randomly.

[3] A number of other values for the missing features weight were tested as well.

Results using this 1-nearest neighbor CBL algorithm for relative pronoun disambiguation using the baseline case representation are shown in Table 2. For these experiments, we drew training and test cases (241 instances) from MUC-3 texts that describe Latin American terrorist events (Chinchor et al., 1993). As above, all results are 10-fold cross validation averages and the parser used to generate training and test cases was the CIRCUS system. The performance of the CBL algorithm is compared to that of: (1) a default rule that always chooses the most recent phrase as the antecedent, and (2) a set of hand-coded heuristics developed for the same task specifically for use in the terrorism domain. Chi-square significance tests indicate: (1) that the hand-coded heuristics perform better (at the 95% level) than the default rule and (2) that the CBL system is not significantly different from either the default rule or the hand-coded heuristics.

Table 2: **Relative Pronoun Disambiguation Using CBL Without Feature Set Selection.** (% correct)

CBL Algorithm w/o feature set selection	**Default Strategy**	**Hand-Coded Heuristics**
76.2	74.3	80.5

In the sections below, we describe the recency bias, the restricted memory bias, and the subject accessibility bias in turn. We show how each bias can be used to automatically modify the baseline case representation and measure the effects of those modifications on the learning algorithm's ability to predict relative pronoun antecedents. Experiments will show that the changes in representation engender a 21.7% increase in accuracy, raising the performance of the CBL algorithm from 69.2% correct to 84.2%. In all experiments below, the same ten training and test set combinations as in the baseline experiments of Table 2 will be used. This procedure ensures that differences in performance are not attributable to the random partitions chosen for the test set.

Incorporating the Recency Bias

In processing language, people consistently show a bias towards the use of the most recent information (e.g., Frazier and Fodor (1978), Gibson (1990), Kimball (1973), Nicol (1988)). In particular, Cuetos and Mitchell (1988), Frazier and Fodor (1978), and others have investigated the importance of recency in finding the antecedents of relative pronouns. They found that there is a preference for choosing the most recent noun phrase in sentences of the form NP V NP OF-PP, with ambiguous relative pronoun antecedents, e.g.:

The journalist interviewed the daughter of the colonel who had had the accident.

In addition, Gibson et al. (1993) looked at phrases of the form: NP1 PREP NP2 OF NP3 RELATIVE-CLAUSE,. E.g.,

...the lamps near the paintings of *the house* that was damaged in the flood.
...the lamps near *the painting* of the houses that was damaged in the flood.
...*the lamp* near the paintings of the houses that was damaged in the flood.

He found that the most recent noun phrase (NP3) was initially preferred as the antecedent and that recognizing antecedents in the NP2 and NP1 positions were significantly harder than recognizing the most recent noun phrase as the antecedent.

We translate this *recency bias* into representational changes for the training and problem cases in two ways. The first is a direct modification to the attributes that comprise the case representation, and the second modifies the weights to indicate a constituent's distance from the relative pronoun.

In the first approach, we label the each constituent feature by its position *relative to the relative pronoun*. This establishes a right-to-left labeling of constituents rather than the left-to-right labeling that the baseline representation incorporates. In Figure 3, for example, "in Congress" receives the attribute *pp1* in the right-to-left labeling because it is a prepositional phrase one position to the left of "who." Similarly, "the hardliners" receives the attribute *np2* because it is a noun phrase two positions to the left of "who." The right-to-left ordering yields a different feature set and, hence, a different case representation. For example, the right-to-left labeling assigns the same antecedent value (i.e., *pp2*) to both of the following sentences:

- "it was a message from *the hardliners* in Congress, who..."
- "it was from *the hardliners* in Congress, who..."

The baseline (left-to-right) representation, on the other hand, labels the antecedents with distinct attributes — *do-pp1* and *v-pp1*, respectively.

In the second approach to incorporating the recency bias, we increment the weight associated with each constituent as a function of its proximity to the relative pronoun (see Table 3). The feature associated with the constituent farthest from the relative pronoun receives a weight of one, and the weights are increased by one for each subsequent constituent. All features added to the case as a result of feature normalization (not shown in Table 3) receive a weight of one.

Table 3: **Incorporating the Recency Bias by Modifying the Weight Vector.**

Phrase	Feature	Baseline weight	Recency weight
It	s	1	1
was	v	1	2
the hardliners	do	1	3
in Congress	do-pp1	1	4
who...			

The results of experiments that use each of the recency representations separately and in a combined form are shown in Table 4. To combine the two implementations of the recency bias, we first relabel the attributes of a case using the right-to-left labeling and then initialize the weight vector using the recency weighting procedure described above. The table shows that the recency weighting representation alone tends to degrade prediction of relative pronoun antecedents as compared to the baseline CBL system. Both the right-to-left labeling and combined representations improve performance — they perform significantly better than the default heuristic, but do not yet exceed the level of the hand-coded heuristics. The final row of results will be described below.

As shown in Table 4, the combined recency bias outperforms the right-to-left labeling despite the fact that the recency weighting tends to lower the accuracy of relative pronoun antecedent prediction when used alone. The right-to-left labeling appears to provide a representation of the local context of the relative pronoun that is critical for finding antecedents. The disappointing perfor-

Sentence:	[It] [was] [the hardliners] [in Congress] [,] who ...
baseline representation:	*(s entity) (v exists) (do human) (do-pp1 entity) (prev1-syntactic-type prep-phrase)* ... (class do)
right-to-left labeling:	*(s entity) (v exists) (np2 human) (pp1 entity) (prev1-syntactic-type prep-phrase)* ... (class np2)

Figure 3: **Incorporating the Recency Bias Using a Right-to-Left Labeling.**

Table 4: **Results for the Recency Bias Representations.**

Case Representation	% Correct
Baseline Representation (no feature selection)	76.2
R-to-L Labeling	79.2
Recency Weighting	75.8
R-to-L + RecWt	80.0
Hand-Coded Heuristics	80.5
Default Heuristic	74.3
Baseline Representation w/o built-in recency bias	69.2

mance of the recency weighting representation, on the other hand, may be caused by (1) its lack of such a representation of local context, and (2) its bias against antecedents that are distant from the relative pronoun (e.g., "...to help especially *those people* living in the Patagonia region of Argentina, who are being treated inhumanely..."). Nineteen of the 241 cases have antecedents that include the often distant subject of the preceding clause.

Furthermore, the recency bias performs well in spite of the fact that the baseline representation already provides a built-in recency bias. The baseline represents the constituent that precedes the relative pronoun up to three times in the baseline representation — as a constituent feature (e.g., "direct object") and via the "last constituent" global context features.[4] The last row in Table 4 shows the performance of the baseline representation when this built-in bias is removed by discarding the *last-constituent* features.

Incorporating the Restricted Memory Bias

Psychological studies have determined that people can remember at most seven plus or minus two items at any one time (Miller, 1956). More recently, Daneman and Carpenter (1983, 1980) show that working memory capacity affects a subject's ability to find the referents of pronouns over varying distances. King and Just (1991) show that differences in working memory capacity can cause differences in the reading time and comprehension of certain classes of relative clauses. Moreover, it has been hypothesized that language learning in humans is successful precisely because limits on information processing capacities allow children to ignore much of the linguistic data they receive (Newport, 1990). Some computational language learning systems (e.g., Elman (1990)) actually build a short term memory directly into the architecture of the system.

Our baseline case representation does not necessarily make use of this *restricted memory bias*, however. Each case is described in terms of the normalized feature set, which contains an average of 38.8 features. Unfortunately, incorporating the restricted memory limitations into the case representation is problematic. Previous restricted memory studies (e.g., short term memory studies) do not state explicitly what the memory limit should be — it varies from five to nine depending on the cognitive task and depending on the size and type of the "chunks" that have to be remembered. In addition, the restricted memory bias alone does not state which chunks, or features, to keep and which to discard.

To apply the restricted memory bias to the baseline case representation, we let n represent the memory limit and, in each of five runs, set n to one of five, six, seven, eight, or nine. Then, for each test case, the system randomly chooses n features from the normalized feature set, sets the weights associated with those features to one, and sets the remaining weights to zero. This effectively dis-

[4]This means that when the constituent immediately preceding "who" in the problem case and a training case match, that constituent accounts for a greater percentage of the similarity score than does any other constituent.

Table 5: **Results for the Restricted Memory Bias Representation.** (% correct, *'s indicate significance with respect to the original baseline result shown in **boldface**, $* \rightarrow p = 0.05$)

Memory Limit	Baseline	R-to-L + RecWt
none	**76.2**	80.0
9	78.3	81.2*
8	74.2	81.2*
7	76.2	80.0
6	75.8	80.4
5	75.0	81.7*

cards all but the n selected features from the case representation. Results for the restricted memory bias representation are shown in Table 5. The first column of results shows the effect of memory limitations on the baseline representation. In general, the restricted memory bias with random feature selection degrades the ability of the system to predict relative pronoun antecedents although none of the changes is statistically significant. This is not surprising given that the current implementation of the bias is likely to discard relevant features as well as irrelevant features. We expect that this bias will have a positive impact on performance only when it is combined with linguistic biases that provide feature relevancy information. This is, in fact, the case: the final column in Table 5 shows the effect of restricted memory limitations on the combined recency representation. To incorporate the restricted memory bias and the combined recency bias into the baseline case representation, we (1) apply the right-to-left labeling, (2) rank the features of the case according to the recency weighting, and (3) keep the n features with the highest weights (where n is the memory limit). Ties are broken randomly.

We expected the merged representation to perform rather well because the combined recency bias representation worked well on its own and because the restricted memory (RM) bias essentially discards features that are distant from the relative pronoun and rarely included in the antecedent. As shown in the last column of Table 5, four out of five RM/recency variations posted higher accuracies than the combined recency representation. In fact, three of the RM/recency representations now outperform the original baseline representation (shown in boldface) at the 95% significance level. (Until this point, the best representation had been the combined recency representation, which significantly outperformed the default heuristic, but not the baseline case representation.)

Incorporating the Subject Accessibility Bias

A number of studies in psycholinguistics have noted the special importance of the first item mentioned in a sentence. In particular, it has been shown that the accessibility of the first discourse object, which very often corresponds to the subject of the sentence, remains high even at the end of a sentence (Gernsbacher et al., 1989). This *subject accessibility bias* is an example of a more general *focus of attention bias*. In vision learning problems, for example, the brightest object in view may be a highly accessible object for the learning agent; in aural tasks, very loud or high-pitched sounds may be highly accessible. We incorporate the subject accessibility bias into the baseline representation by increasing the weight associated with the constituent attribute that represents the subject of the clause preceding the relative pronoun whenever that feature is part of the normalized feature set.

Table 6: **Results for the Subject Accessibility Bias Representation.** (% correct)

Baseline	76.2
Baseline, SubjWt=2	75.0
Baseline, SubjWt=5	74.2
Baseline, SubjWt=7	73.7
Baseline, SubjWt=10	73.3

Table 6 shows the effects of allowing matches on the subject attribute to contribute two, five, seven, and ten times as much as they did in the baseline representation. The weights were chosen more or less arbitrarily. Results indicate that incorporation of the subject accessibility bias never improves performance of the learning algorithm, although dips in performance are never statistically significant. At first it may seem surprising that this bias does not result in a better representation. Like the recency bias, however, the baseline representation already encodes the subject accessibility bias by explicitly recognizing the subject as a major constituent of the sentence (i.e., "s") rather than by labeling it merely as a low-level noun phrase (i.e., "np"). It may be that this built-in encoding of the bias is adequate or that, like the restricted memory bias, additional modifications to the baseline representation are required before the subject accessibility bias can have a positive effect on the learning algorithm's ability to find relative pronoun antecedents.

Table 7 shows the effects of merging the subject accessibility bias with both recency biases and the restricted memory bias (RM). The results in the first column (Baseline) are just the results from Table 6 — they indicate the performance of the baseline case representation with various levels of the subject accessibility bias. The second column shows the effect of incorporating the subject accessibility bias into the combined recency bias representation. To create this merged representation, we first establish the right-to-left labeling of features and then add together the weight vectors recommended by the recency weighting and subject accessibility biases. As was the case with the baseline representation, incorporation of the subject accessibility bias steadily decreases performance of the learning algorithm as the weight on the subject constituent is increased. None of the changes is statistically significant.

The remaining five columns of Table 7 show the effects of incorporating all three linguistic biases into the baseline case representation. To create this representation, we (1) relabel the attributes using the right-to-left labeling, (2) incorporate the subject and recency weighting representations by adding the weight vectors proposed by each bias, (3) apply the restricted memory bias by keeping only the n features with the highest weights (where n is the memory limit) and choosing randomly in case of ties. Results for these experiments indicate that some combinations of the linguistic bias parameters work very well together and others do not. In general, associating a weight of two with the subject constituent improves the accuracy of the learning algorithm as compared to the corresponding representation that omits the subject accessibility bias. (Compare the first and second rows of results). In particular, three representations (shown in italics) now outperform the best previous representation (which had the r-to-l labeling, recency weighting, memory limit = 5 and achieved 81.7% correct). In addition, the best-performing representation now outperforms the hand-coded relative pronoun disambiguation rules (84.2% vs. 80.5%) at the 90% significance level.

In summary, this section presented a linguistic bias approach to feature set selection and applied it to the problem of finding the antecedent of the relative pronoun "who." Our experiments showed that performance of the case-based learning algorithm steadily improved as each of the available linguistic biases was used to modify the baseline case representation. Although one would not expect monotonic improvement to continue forever, it is clear that explicit incorporation of linguistic biases into the case representation can improve the learning algorithm performance for the relative pronoun disambiguation task. Table 8 summarizes these results. When all three biases are included in the case representation, the learning algorithm performs significantly better than the hand-coded rules (84.2% correct vs. 80.5% correct) at the 90% confidence level.

Discussion and Conclusions

It should be emphasized that modifications to the baseline case representation in response to each of the individual linguistic biases are performed **automatically** by the CBL system, subject to the constraints provided in Table 9. Upon invocation of the CBL algorithm, the user need only specify (1) the names of the biases to incorporate into the case representation, and (2) any parameters required for those biases (e.g., the memory limit for the restricted memory bias).

In addition, the linguistic bias approach to feature set selection relies on the following general procedure when incorporating more than one linguistic bias into the baseline representation:

1. First, incorporate any bias that relabels attributes (e.g., r-to-l labeling).
2. Then, incorporate biases that modify feature weights by adding the weight vectors proposed by each bias (e.g., recency weighting, subject accessibility bias).
3. Finally, incorporate biases that discard features (e.g., restricted memory bias), but give preference to those features assigned the highest weights in Step 2.

Thus far, we have implemented just three linguistic biases, all of which represent broadly applicable cognitive processing limitations. We expect that additional biases will be needed to handle new natural language learning tasks, but that, in general, a relatively small set of linguistic biases should be adequate for handling large number of problems in natural language learning. Examples of other useful linguistic biases to make available include: minimal attachment, right association, lexical preference biases, and a syntactic structure identity bias.

One important problem that we have not addressed is how to select automatically the combination of linguistic biases that will achieve the

Table 7: **Additional Results for the Subject Accessibility Bias Representation.** (% correct, *'s indicate significance with respect to the original baseline result shown in **boldface**, $* \rightarrow p = 0.05$, $** \rightarrow p = 0.01$; RM refers to the memory limit).

Subject Weight	**Baseline**	**SubjAcc R-to-L RecWt**	**SubjAcc R-to-L RecWt RM=5**	**SubjAcc R-to-L RecWt RM=6**	**SubjAcc R-to-L RecWt RM=7**	**SubjAcc R-to-L RecWt RM=8**	**SubjAcc R-to-L RecWt RM=9**
none	**76.2**	80.0	81.7*	80.4	80.0	81.2*	81.2*
2	75.0	79.6	*84.2***	*82.5**	*82.1**	81.2*	80.8
5	74.2	78.3	79.6	79.2	78.3	80.4	79.6
7	73.7	77.5	79.6	79.2	77.9	76.7	77.9
10	73.3	76.7	79.6	79.2	78.3	80.4	79.6

Table 8: **Summary of Linguistic Bias Results.**

Case Representation	**% Correct**
Baseline w/o Built-in Recency Bias	69.2
Default Heuristic: Choose Most Recent Phrase	74.3
Baseline	76.2
Baseline + Recency Bias	80.0
Hand-Coded Heuristics	80.5
Baseline + Recency Bias + Restricted Memory Bias (limit=5)	81.7
Baseline + Recency Bias + Restricted Memory Bias (limit=5) + Subject Accessibility Bias (subj wt=2)	84.2

Table 9: **Linguistic Bias Modifications.**

Bias	**Assumptions**	**Parameters**
Recency (r-to-l labeling)	Attribute names indicate recency	Function mapping original attribute names to new attribute names
Recency (recency weighting)	Attributes in original case are provided in inverse recency order	None
Restricted Memory	None	memory limit
Focus of Attention (subject accessibility)	None	Weight factor, attribute associated with object of focus, e.g., the subject

best performance for a particular natural language learning task. Our current approach assumes that the expert knowledge of computational linguists is easier to apply at the level of linguistic bias selection than at the feature set selection level — so at the very least, this expert knowledge can be used to seed the bias selection algorithm. For the relative pronoun task, for example, we assumed that all three linguistic biases were relevant and then exhaustively enumerated all combinations of the biases, choosing the combination that performed best in cross-validation testing. Because this method will get quickly out of hand as additional biases are included or parameters tested, future work should investigate less costly alternatives to linguistic bias selection.

In addition, we have tested the linguistic bias approach to feature selection on just one natural language learning task. We believe, however, that it offers a general approach for case-based learning of natural language. In theory, it allows system developers to use the same underlying case representation for a variety of problems in NLP rather than developing a new representation as each new task is tackled. The underlying case representation only has to change when new knowledge sources become available to the NLP system in which the CBL system is embedded. Hence, the baseline case representation is parser-dependent (i.e., NLP system-dependent) rather than task-dependent.

In particular, we are currently applying the linguistic bias CBL approach to the problem of general pronoun resolution. While it appears that our existing linguistic bias set will be of use, we believe that the CBL system will benefit from additional linguistic biases. Centering constraints (see Brennan et al., 1987), for example, can be encoded as linguistic biases and applied to the pronoun resolution task to increase system performance .

Furthermore, we have focused on applying the linguistic bias approach to feature set selection for case-based learning algorithms only. In future work, we plan to investigate the use of the approach for feature selection in conjunction with other standard machine learning algorithms. Here we expect that very different manipulations of the baseline case representation will be needed to implement the linguistic biases presented in this paper.

Finally, the viability of both the linguistic bias approach to feature set selection and the general CBL approach to natural language learning must be tested using much larger corpora. Experiments on case-based part-of-speech tagging by researchers at Tilburg University (Daelemans et al., submitted), however, indicate that the CBL approach to natural language learning will scale to much larger data sets.

In summary, this paper begins to address the issue of "algorithm vs. representation" for case-based learning of linguistic knowledge. We have shown empirically that the feature set used to describe training and test instances plays an important role for a number of tasks in natural language understanding. In addition, we have presented an automated approach to feature set selection for case-based learning of linguistic knowledge. The approach takes a baseline case representation and modifies it in response to one of three linguistic biases by adding, deleting, and weighting features appropriately. We applied the technique to the task of relative pronoun disambiguation and found that the case-based learning algorithm improves as relevant biases are used to modify the underlying case representation. Finally, we have argued that the linguistic bias approach to feature set selection offers new possibilities for case-based learning of natural language. It simplifies the process of designing an appropriate instance representation for individual natural language learning tasks because system developers can safely include in the baseline instance representation features for all available knowledge sources. In the long run, it may obviate the need for separate instance representations for each linguistic knowledge acquisition task. More importantly, the linguistic bias CBL approach to natural language learning offers a mechanism for explicitly combining the frequency information available from corpus-based techniques with linguistic bias information employed in traditional linguistic and knowledge-based approaches to natural language processing.

References

(Aha et al., 1991) D. Aha, D. Kibler, and M. Albert. 1991. Instance-Based Learning Algorithms. *Machine Learning*, 6(1):37–66.

(Aha, 1989) D. Aha. 1989. Instance-Based Learning Algorithms. In *Proceedings of the Sixth International Conference on Machine Learning*, pages 387–391, Cornell University, Ithaca, NY. Morgan Kaufmann.

(Almuallim and Dietterich, 1991) H. Almuallim and T. G. Dietterich. 1991. Learning With Many Irrelevant Features. In *Proceedings of the Ninth National Conference on Artificial Intelligence*, pages 547–552, Anaheim, CA. AAAI Press / MIT Press.

(Aone and Bennett, 1995) Chinatsu Aone and William Bennett. 1995. Evaluating Automated and Manual Acquisition of Anaphora Resolution Strategies. In *Proceedings of the 33rd*

Annual Meeting of the ACL, pages 122–129. Association for Computational Linguistics.

(Bosch and Daelemans, 1993) A. van den Bosch and W. Daelemans. 1993. Data-oriented methods for grapheme-to-phoneme conversion. In *Proceedings of European Chapter of ACL*, pages 45–53, Utrecht. Also available as ITK Research Report 42.

(Brennan et al., 1987) Susan E. Brennan, Marilyn Walker Friedman, and Carl J. Pollard. 1987. A Centering Approach to Pronouns. In *Proceedings of the 25th Annual Meeting of the ACL*. Association for Computational Linguistics.

(Cardie and Lehnert, 1991) C. Cardie and W. Lehnert. 1991. A Cognitively Plausible Approach to Understanding Complicated Syntax. In *Proceedings of the Ninth National Conference on Artificial Intelligence*, pages 117–124, Anaheim, CA. AAAI Press / MIT Press.

(Cardie, 1992a) C. Cardie. 1992a. Corpus-Based Acquisition of Relative Pronoun Disambiguation Heuristics. In *Proceedings of the 30th Annual Meeting of the ACL*, pages 216–223, University of Delaware, Newark, DE. Association for Computational Linguistics.

(Cardie, 1992b) C. Cardie. 1992b. Learning to Disambiguate Relative Pronouns. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, pages 38–43, San Jose, CA. AAAI Press / MIT Press.

(Cardie, 1993a) C. Cardie. 1993a. A Case-Based Approach to Knowledge Acquisition for Domain-Specific Sentence Analysis. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pages 798–803, Washington, DC. AAAI Press / MIT Press.

(Cardie, 1993b) C. Cardie. 1993b. Using Decision Trees to Improve Case-Based Learning. In P. Utgoff, editor, *Proceedings of the Tenth International Conference on Machine Learning*, pages 25–32, University of Massachusetts, Amherst, MA. Morgan Kaufmann.

(Cardie, 1994) C. Cardie. 1994. *Domain-Specific Knowledge Acquisition for Conceptual Sentence Analysis*. Ph.D. thesis, University of Massachusetts, Amherst, MA. Available as University of Massachusetts, CMPSCI Technical Report 94–74.

(Chinchor et al., 1993) N. Chinchor, L. Hirschman, and D. Lewis. 1993. Evaluating Message Understanding Systems: An Analysis of the Third Message Undestanding Conference (MUC-3). *Computational Linguistics*, 19(3):409–449.

(Cuetos and Mitchell, 1988) F. Cuetos and D. C. Mitchell. 1988. Cross-Linguistic Differences in Parsing: Restrictions on the Use of the Late Closure Strategy in Spanish. *Cognition*, 30(1):73–105.

(Daelemans et al., 1994) W. Daelemans, G. Durieux, and S. Gillis. 1994. The Acquisition of Stress: A Data-Oriented Approach. *Computational Linguistics*, 20(3):421–451.

(Daelemans et al., submitted) W. Daelemans, J. Zavrel, Berck P., and Gillis S. submitted. Memory-Based Part of Speech Tagging. Tilburg University.

(Daneman and Carpenter, 1980) M. Daneman and P. A. Carpenter. 1980. Individual Differences in Working Memory and Reading. *Journal of Verbal Learning and Verbal Behavior*, 19:450–466.

(Daneman and Carpenter, 1983) M. Daneman and P. A. Carpenter. 1983. Individual Differences in Integrating Information Between and Within Sentences. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 9:561–584.

(Elman, 1990) J. Elman. 1990. Finding Structure in Time. *Cognitive Science*, 14:179–211.

(Frazier and Fodor, 1978) L. Frazier and J. D. Fodor. 1978. The Sausage Machine: A New Two-Stage Parsing Model. *Cognition*, 6:291–325.

(Gernsbacher et al., 1989) M. A. Gernsbacher, D. J. Hargreaves, and M. Beeman. 1989. Building and Accessing Clausal Representations: The Advantage of First Mention Versus the Advantage of Clause Recency. *Journal of Memory and Language*, 28:735–755.

(Gibson et al., 1993) E. Gibson, N. Pearlmutter, E. Canseco-Gonzalez, and G. Hickok. 1993. Cross-linguistic Attachment Preferences: Evidence from English and Spanish. In *Sixth Annual CUNY Sentence Processing Conference*, University of Massachusetts, Amherst, MA. Only abstract in the Sentence Processing Conference proceedings. Full manuscript to appear in journal.

(Gibson, 1990) E. Gibson. 1990. Recency Preferences and Garden-Path Effects. In *Proceedings of the Twelfth Annual Conference of the Cognitive Science Society*, Massachusetts Institute of Technology, Cambridge, MA. Lawrence Erlbaum Associates.

(Kimball, 1973) J. Kimball. 1973. Seven Principles of Surface Structure Parsing in Natural Language. *Cognition*, 2:15–47.

(King and Just, 1991) J. King and M. A. Just. 1991. Individual Differences in Syntactic Processing: The Role of Working Memory. *Journal of Memory and Language*, 30:580–602.

(Kolodner, 1993) J. Kolodner. 1993. *Case-Based Reasoning.* Morgan Kaufmann, San Mateo, CA.

(Langley and Sage, in press) P. Langley and S. Sage. in press. Scaling to domains with irrelevant features. In R. Greiner, editor, *Computational learning theory and natural learning systems*, volume 4. The MIT Press, Cambridge, MA.

(Lehnert, 1990) W. Lehnert. 1990. Symbolic/Subsymbolic Sentence Analysis: Exploiting the Best of Two Worlds. In J. Barnden and J. Pollack, editors, *Advances in Connectionist and Neural Computation Theory*, pages 135–164. Ablex Publishers, Norwood, NJ.

(Litman, 1994) Diane J. Litman. 1994. Classifying Cue Phrases in Text and Speech Using Machine Learning. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, pages 806–813. AAAI Press / MIT Press.

(Magerman, 1995) David M. Magerman. 1995. Statistical Decision-Tree Models for Parsing. In *Proceedings of the 33rd Annual Meeting of the ACL*, pages 276–283. Association for Computational Linguistics.

(McCarthy and Lehnert, 1995) Joseph F. McCarthy and Wendy G. Lehnert. 1995. Using Decision Trees for Coreference Resolution. In C. Mellish, editor, *Proceedings of the Fourteenth International Conference on Artificial Intelligence*, pages 1050–1055.

(Miller, 1956) G. A. Miller. 1956. The Magical Number Seven, Plus or Minus Two: Some Limits on our Capacity for Processing Information. *Psychological Review*, 63(1):81–97.

(MUC, 1994) 1994. *Proceedings of the Fifth Message Understanding Conference (MUC-5).* Morgan Kaufmann, San Mateo, CA.

(Newport, 1990) E. Newport. 1990. Maturational Constraints on Language Learning. *Cognitive Science*, 14:11–28.

(Nicol, 1988) J. Nicol. 1988. *Coreference Processing During Sentence Comprehension.* Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA.

(Pereira, 1994) F. Pereira. 1994. Frequencies vs Biases: Machine learning problems in natural language processing. In *Proceedings of the Eleventh International Conference on Machine Learning*, page 380, Rutgers University, New Brunswick, NJ. Morgan Kaufmann.

(Quinlan, 1983) J. R. Quinlan. 1983. Learning Efficient Classification Procedures and Their Application to Chess End Games. In R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, editors, *Machine Learning: An Artificial Intelligence Approach.* Morgan Kaufmann, San Mateo, CA.

(Quinlan, 1992) J. R. Quinlan. 1992. *C4.5: Programs for Machine Learning.* Morgan Kaufmann, San Mateo, CA.

(Riesbeck and Schank, 1989) C. Riesbeck and R. Schank. 1989. *Inside Case-Based Reasoning.* Erlbaum, Northvale, NJ.

(Riloff and Lehnert, 1994) E. Riloff and W. Lehnert. 1994. Information extraction as a basis for high-precision text classification. *ACM Transactions on Information Systems*, 12(3):296–333.

(Simmons and Yu, 1992) Robert F. Simmons and Yeong-Ho Yu. 1992. The Acquisition and Use of Context-Dependent Grammars for English. *Computational Linguistics*, 18(4):391–418.

(Stanfill and Waltz, 1986) C. Stanfill and D. Waltz. 1986. Toward Memory-based Reasoning. *Communications of the ACM*, 29:1213–1228.

(Zelle and Mooney, 1993) J. Zelle and R. Mooney. 1993. Learning Semantic Grammars with Constructive Inductive Logic Programming. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pages 817–822, Washington, DC. AAAI Press / MIT Press.

(Zelle and Mooney, 1994) J. Zelle and R. Mooney. 1994. Inducing Deterministic Prolog Parsers from Treebanks: A Machine Learning Approach. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, pages 748–753, Seattle, WA. AAAI Press / MIT Press.