Machine Learning, 41, 85–116, 2000

A Cognitive Bias Approach to Feature Selection and Weighting for Case-Based Learners

CLAIRE CARDIE cardie@cs.cornell.edu
Department of Computer Science, Cornell University, Ithaca, NY 14853–7501, USA

Editor: David W. Aha

Abstract. Research in psychology, psycholinguistics, and cognitive science has discovered and examined numerous psychological constraints on human information processing. Short term memory limitations, a focus of attention bias, and a preference for the use of temporally recent information are three examples. This paper shows that psychological constraints such as these can be used effectively as domain-independent sources of bias to guide feature set selection and weighting for case-based learning algorithms.

We first show that cognitive biases can be automatically and explicitly encoded into the baseline instance representation: each bias modifies the representation by changing features, deleting features, or modifying feature weights. Next, we investigate the related problems of cognitive bias selection and cognitive bias interaction for the feature weighting approach. In particular, we compare two cross-validation algorithms for bias selection that make different assumptions about the independence of individual component biases. In evaluations on four natural language learning tasks, we show that the bias selection algorithms can determine which cognitive bias or biases are relevant for each learning task and that the accuracy of the case-based learning algorithm improves significantly when the selected bias(es) are incorporated into the baseline instance representation.

Keywords: case-based learning, instance-based learning, feature set selection, feature weighting, natural language learning

1. Introduction

Inductive concept acquisition has always been a primary interest for researchers in the field of machine learning (Langley, 1996; Mitchell, 1997). Independently, psychologists, psycholinguists, and cognitive scientists have examined the effects of numerous psychological limitations on human information processing (Wilson & Keil, 1999). However, despite the fact that concept learning is a basic cognitive task, cognitive processing limitations are rarely exploited in the design of machine learning systems for concept acquisition.

This paper shows that cognitive processing limitations can be used effectively as domain-independent sources of bias to guide feature set selection and, as a result, to improve learning algorithm performance. We first describe how cognitive biases can be automatically and explicitly encoded into a training instance representation. In particular, we use a simple case-based learning algorithm (*k*-nearest neighbor (Cover & Hart, 1967)) and initially focus on a single learning task from the field of natural language processing.

After presenting a baseline instance representation for the task, we modify the representation in response to three cognitive biases—a focus of attention bias (Broadbent, 1958), a recency bias (Kimball, 1973), and short term memory limitations (Miller, 1956). In a series of experiments, we compare the modified instance representations to the baseline description and find that, when used in isolation, only one cognitive bias significantly improves system performance. We hypothesize that additional gains in accuracy might be achieved by applying two or more cognitive biases simultaneously to the baseline instance representation.

As more psychological processing limitations are included in the instance representation, however, the system must address the related issues of cognitive bias interaction and cognitive bias selection. As a result, the paper next presents two methods for cognitive bias selection that make varying assumptions about the independence of individual processing limitations. In general, each method combines search and cross validation. The greedy approach to bias selection incrementally incorporates into the baseline representation the best-performing individual cognitive biases while the learning algorithm's accuracy continues to improve. This method assumes that there will be few deleterious bias interactions. In contrast, the second algorithm for bias selection makes no assumptions about bias interactions and instead exhaustively evaluates all combinations of the available cognitive biases. Our work differs from most previous work in that we use search and cross validation for bias selection rather than for feature selection; the selected biases are then responsible for directing feature set selection and feature weighting. The results of our experiments show that the bias selection algorithms can determine which cognitive biases are relevant for each learning task and that performance of the case-based learning algorithm improves significantly when the selected bias or biases are incorporated into the baseline instance representation.

Finally, we investigate the generality of the cognitive bias approach to feature set selection. We first show that two additional cognitive biases can be translated into representational changes for the baseline instance representation. We then apply the feature selection algorithm with all five cognitive biases to three additional natural language learning tasks. Again, we find that (1) the cognitive bias selection algorithms are able to choose one or more appropriate biases for each task, and (2) the incorporation of these relevant biases significantly improves the learning algorithm's performance.

The remainder of the paper is organized as follows. The next two sections describe the first natural language learning task—relative pronoun disambiguation—and its baseline instance representation. This task will be used throughout the paper to introduce components of the cognitive bias approach to feature set selection. Section 4 presents the case-based learning algorithm and its evaluation on the relative pronoun task. The method used to incorporate independently each of the three primary cognitive biases—focus of attention, recency, short term memory limitations—is described in Section 5. Section 6 proposes and evaluates the alternative approaches to bias selection briefly outlined above. An evaluation of the cognitive bias approach to feature set selection on additional data sets comprises Section 7. Cardie (1999) describes the implications of this work for natural language processing rather than machine learning.

2. Relative pronoun disambiguation

The goal of the machine learning algorithm for the first natural language task is to disambiguate wh-words (e.g., who, which, where) in sentences like:

Tony saw *the boy* **who** won the award.

In particular, the learning algorithm must locate the phrase or phrases, if any, that represent the antecedent of the wh-word given a description of the context in which the wh-word occurs. For the sample sentence, the system should recognize that "the boy" is the antecedent of "who" because "who" refers to "the boy." Finding the antecedents of relative pronouns is a crucial task for natural language understanding systems in part because the antecedent fills a semantic role in two clauses. In the sample sentence, for example, "the boy" is both the object of "saw"and the implicit actor of "won."

In addition, we focus on disambiguation of relative pronouns because (1) they occur frequently in the long, multi-clause sentences of many real-world texts, (2) disambiguation of relative pronouns was determined to be critical for the larger information extraction task within which the learning algorithm was embedded, (3) our existing natural language processing (NLP) system (Lehnert et al., 1991) included hand-crafted disambiguation heuristics with which we could directly compare the learning algorithm performance, and (4) there is a large body of literature on the human processing of relative clauses. (Both the corpus and the broader information extraction task will be described in Section 3.) We focus more specifically on learning disambiguation heuristics for "who" because this was the most frequent relative pronoun to appear in the corpus, occurring in about one out of every ten sentences and at a higher frequency for the most important documents, i.e., for texts that are actually relevant to the information extraction task. In addition, the majority of psycholinguistic studies of human processing of relative clauses focus on "who."

Although finding relative pronoun antecedents seems a simple enough task, there are many factors that make it difficult:

The head noun of the antecedent of a relative pronoun does not appear in a consistent syntactic constituent or position. In both examples S1 and S2 of figure 1, for example, the antecedent is "the boy." In S1, however, "the boy" is the direct object of the preceding clause, while in S2 it appears as the subject. On the other hand, the head of the antecedent is the phrase that immediately precedes "who" in both cases. S3, however, shows that this is not always the case. In fact, the antecedent head may be very distant from the relative pronoun (e.g., S4).

The antecedent may be a conjoined noun phrase. In S5, for example, the antecedent of "who" is a conjunction of three phrases.

There may be more than one semantically valid antecedent. In S6, "GE" refers to the same entity as "our sponsor." As a result, the antecedent of "who" can be either "our sponsor," "GE," or the entire phrase "our sponsor, GE." A similar situation occurs in predicate nominative constructions.

S1.	Tony saw *the boy* **who** won the award.
S2.	*The boy* **who** gave me the book had red hair.
S3.	Tony ate dinner with *the men* from Detroit **who** sold computers.
S4.	I spoke to *the woman* with the black shirt and green hat over in the far corner of the room **who** wanted a second interview.
S5.	I'd like to thank *Jim, Terry, and Shawn,* **who** provided the desserts.
S6.	I'd like to thank *our sponsor, GE,* **who** provided financial support.
S7.	We wondered **who** stole the watch.
S8.	*The woman* from Philadelphia **who** played soccer was my sister.
S9.	The gifts from *the children* **who** attended the party are on the table.

Figure 1. Antecedents of "who".

Sometimes there is no apparent antecedent. As in S7, sentence analyzers must be able to distinguish uses of "who" that have no antecedent (e.g., interrogatives) from instances of true relative pronouns.

Locating the antecedent requires the assimilation of both syntactic and semantic knowledge. The syntactic structure of the clause preceding "who" in sentences S8 and S9, for example, is identical. The antecedent in each case is different, however. In S8, the antecedent is the subject, "the woman." In S9, it is the head noun of the prepositional phrase, i.e., "the children."

Despite these difficulties, we will show that a machine learning system can learn to locate the antecedent of "who" given a description of the clause that precedes it.

3. The baseline instance representation

In all experiments, we use the CIRCUS sentence analyzer (Lehnert, 1990) to generate training instances. The system generates one instance for every occurrence of "who" that appears in texts taken from the MUC terrorism corpus. This corpus was developed in conjunction with the third Message Understanding Conference (MUC-3, 1991), a performance evaluation of state-of-the-art information extraction systems. In general, an information extraction system takes as input an unrestricted text and "summarizes" the text with respect to a prespecified topic or domain of interest: it finds useful information about the domain and encodes that information in a structured, template format, suitable for populating databases (Cardie, 1997). The CIRCUS system has been a consistently strong performer in the MUC evaluations. The MUC collection consists of 1300 documents including newswire stories, speeches, radio and TV broadcasts, interviews, and rebel communiques. Texts contain both well-formed and ungrammatical sentences; all texts are entirely in upper case.

Each training instance for the relative pronoun task is a list of attribute-value pairs that encode the context in which the wh-word is found. In addition, each training instance is also annotated with a class value that describes the position of the correct antecedent for "who" in each example. This antecedent class value is the feature to be predicted by the learning algorithm during testing. The details of the baseline instance representation depend, in part,

Table 1. Relative pronoun resolution baseline instance representation for: "*The man from Oklahoma, who . . .*"

Phrases	Features		
The man	(S	*t*)	
	(S-SEM	*human*)	
from Oklahoma	(S-PP-1	*t*)	
	(S-PP-1-SEM	*location*)	
	(S-PP-1-PREP	*from*)	
,	(S-PP-1-MARK	*comma*)	
who . . .	(PREV-TYPE	*comma*)	Antecedent: (S)

on the key characteristics of CIRCUS's sentence analyzer:

- The CIRCUS parser recognizes phrases as it finds them in its left-to-right traversal of a sentence.
- It recognizes major constituents like the subject, verb, and direct object.
- It makes no immediate decisions on structural attachment. In particular, it does not handle conjunctions, appositives, or prepositional phrase attachment.
- CIRCUS uses one or more semantic features to describe every noun and adjective in its lexicon.[1] For example, "mayor" is *human*; "ELN" is an *organization*; and the noun "murder" is an *attack*.
- CIRCUS keeps track of the most recently recognized entity, e.g., word, phrase, punctuation mark.

Relative pronoun disambiguation cases are comprised of five types of features: CONSTITUENT, SEM, PREP, MARK, and PREV-TYPE. Because the antecedent of "who" usually appears as one or more phrases in the preceding clause, cases contain one or more attribute-value pairs to describe each phrase in the clause that precedes "who." Consider, for example, the case in Table 1, in which there are two phrases before "who"—"the man" and "from Oklahoma." For each phrase, there is one CONSTITUENT feature that denotes the syntactic class and position of the phrase as it was encountered by the parser. The feature (S *t*) indicates that the parser has recognized "the man" as the subject (S) of the sentence, i.e., there exists a subject. Similarly, (S-PP-1 *t*) declares that "from Oklahoma" is the first prepositional phrase (PP) that follows the subject.

In addition, there is a SEM feature for each phrase that provides its semantic classification if one is available from the system lexicon. The SEM feature for the subject (S-SEM *human*) indicates that "the man" is *human*; "from Oklahoma" specifies a *location*. If the phrase is a prepositional phrase, then we include a PREP feature that denotes the preposition. *From* is the value of this feature for "from Oklahoma." If the phrase is followed by a punctuation mark or conjunction or both, then a MARK feature is included to denote this. The MARK feature for "from Oklahoma" indicates that a *comma* follows the phrase.

Finally, one PREV-TYPE feature per instance denotes the syntactic type of the linguistic entity that immediately precedes the wh-word. In the sample sentence, the value of the

Table 2. Relative pronoun resolution baseline instance representation for: "*I thank Nike and Reebok, who . . .*"

Phrases	Features		
I	(S	*t*)	
	(S-SEM	*pronoun*)	
thank	(V	*t*)	
Nike	(DO	*t*)	
	(DO-SEM	*proper-name*)	
and	(DO-MARK	*and*)	
Reebok	(DO-NP-1	*t*)	
	(DO-NP-1-SEM	*proper-name*)	
,	(DO-NP-1-MARK	*comma*)	
who . . .	(PREV-TYPE	*comma*)	Antecedent: (DO DO-NP-1)

PREV-TYPE feature is *comma*. If no comma had preceded "who," then the value of PREV-TYPE would have been *prepositional-phrase*.

When clauses contain conjunctions and appositives, each phrase in the construct is labeled separately. In the sentence of Table 2, for example, the direct object of "thank" is the conjunction "Nike and Reebok." However, in CIRCUS, and therefore in our instance representation, "Nike" is recognized as the direct object (DO) and "Reebok" as the first noun phrase that follows the direct object (DO-NP-1). Because verb phrases have no semantic features in CIRCUS, there is no SEM feature for verbs. The examples of Tables 1 and 2 also illustrate an important characteristic of the relative pronoun data set: features differ across cases because the structure of the preceding clause varies for each relative pronoun occurrence.

Every training instance is also annotated with class information—a list of the CONSTITUENT attributes that represent the position of the antecedent of "who" or *none* if no antecedent is present. In the sentence of Table 1, for example, the antecedent of "who" is "the man." Because this phrase is represented as the S constituent, the antecedent class value is (S). Sometimes, the antecedent is a conjunction of constituents. In these cases, we represent the antecedent as a list of the CONSTITUENT attributes associated with each element of the conjunction. In the sentence of Table 2, for example, "who" refers to the conjunction "Nike and Reebok;" therefore, the antecedent is encoded as (DO DO-NP-1). Appositive and predicate nominative constructions result in a set of semantically equivalent antecedents, all of which become part of the antecedent class information. In the sentence "I thank our sponsor, GE, who . . . " the antecedent can be "GE" (DO-NP-1), "our sponsor" (DO), or the combined phrase "our sponsor, GE" (DO DO-NP-1). To be considered correct during testing, only one of these three options must be predicted.

4. Evaluation of the baseline instance representation

This section describes the case-based learning algorithm and uses it to evaluate the baseline case representation for relative pronoun disambiguation.

4.1. The case-based learning algorithm

Throughout the paper, we employ a simple case-based, or instance-based, learning algorithm (e.g., Aha, Kibler, & Albert, 1991). During the training phase, all relative pronoun instances are simply stored in a case base. Then, given a new relative pronoun instance, a weighted 1-nearest neighbor (1-nn) case retrieval algorithm predicts its antecedent:

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

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

   where N denotes the test case features, w_{N_i} is the weight of the ith feature in N, X_{N_i} is the value of feature N_i in the test 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. (For the baseline experiments, $w_{N_i} = 1$.)
2. Return the cases with the highest score.
3. If a single case is retrieved, use its antecedent class information to find the antecedent in the test case. Otherwise, let the retrieved cases vote on the position of the antecedent.

If the antecedent of the top-ranked case is DO (direct object), for example, then the direct object of the test case sentence would be selected as the antecedent. Sometimes, however, the retrieved case may list more than one option as the antecedent (for appositive and predicate nominative constructions). In these cases, we choose the first option in the antecedent list whose constituents overlap with those in the current example.

The above case retrieval algorithm matches only on features that appear in the test case. Alternatively, the retrieval algorithm could normalize the feature set across the training cases and then match with respect to this expanded feature set. We obtained comparable performance on the relative pronoun task when using a normalized feature set, but will not discuss those results here.

4.2. The relative pronoun data set

The relative pronoun data set contains 241 instances—CIRCUS generates one case for each occurrence of "who" in 150 texts from the MUC-3 corpus. The correct antecedent for each case must be specified by a human supervisor or by accessing a version of the training corpus that has been annotated with relative clause attachment information. For the experiments in this paper, we made one pass through the data set to correct a small number of obvious parsing and semantic class disambiguation errors.

The performance of the learning algorithm depends, in part, on the underlying characteristics of this data set. First, instances contain between two and 31 features and represent clauses that have from one to 11 phrases. 77% of the cases are unique. In addition, the antecedent class takes on 60 distinct values across the data set. In particular, there are ten

instances with unique antecedent values. This establishes an upper limit of 96% accuracy when using the baseline instance representation.

The data set contains just one pair of 'ambiguous' instances that have the same set of attribute-value pairs, but a different antecedent value. Here our instance representation was too coarse to differentiate wh-word contexts. In particular, it lacked the necessary lexical features to distinguish one type of pronoun from another. More importantly, 51% of the cases involve syntactically ambiguous constructs with respect to relative pronoun resolution. These include sentences like the following, where the head noun of either constituent$_1$ or constituent$_2$ is a syntactically viable antecedent:

I walked [with the man]$_2$ [from Detroit]$_1$ **who** . . .
I saw [the daughter]$_2$ [of the colonel]$_1$ **who** . . .

In theory, the case-based learning algorithm can use the available semantic class information or punctuation to correctly handle most of these cases (e.g. "Detroit" is an unlikely antecedent of **who** since it is a location). Even so, a small number of cases (like the second example) will remain ambiguous without additional context.

4.3. Results

The results for the case-based learning algorithm using the baseline case representation are shown in Table 3. In these runs, all weights w_f are set to 1 and, as mentioned above, the 1-nn algorithm performs calculations using the test case features rather than a normalized feature set. Unless otherwise stated, all results in the paper are leave-one-out cross validation averages and all statements of statistical significance are at or above the 95% confidence level ($p \leq 0.05$). We use both χ^2 and McNemar's tests for statistical significance.

The first row of Table 3 shows the accuracy of this baseline case-based learning algorithm. The remaining rows of the table show the performance of five additional baseline systems. Three systems implement default rules for relative pronoun disambiguation. The first heuristic always chooses the Most Recent Constituent as the antecedent and resorts to *none* if the preceding clause contained no completed constituents. One might expect a system that looks specifically for human antecedents to perform better. Results for the Most Recent Human default rule show that this is not the case. The problem is that many

Table 3. Results for the baseline representation, default rules, hand-coded heuristics, and the IG-CBL feature weighting algorithm for relative pronoun resolution (% correct).

1-nn with baseline instance representation	78.8
Most recent constituent	77.6
Most recent human	58.1
Most recent human or proper name	71.4
Hand-coded heuristics	80.5
IG-CBL	56.8

legitimate antecedents of "who" are characterized by semantic features other than *human*. Unfortunately, looking for more complicated semantic feature combinations like those of the third rule (Most Recent Human or Proper Name) does no better than the simplest default rule.

The fourth baseline system of Table 3 employs a set of hand-crafted heuristics for relative pronoun resolution that were developed for use in the MUC-3 performance evaluation. The heuristics consisted of approximately 30 rules. The following are examples:

> If there is no verb and no subject in the preceding clause, and the last constituent was an NP, then the antecedent is the head of the last constituent.
>
> If there is no verb in the preceding clause, and the token that precedes "who" is a comma, then the antecedent is the head of the subject of the preceding clause.

In general, the rules make use of the same syntactic and semantic information that is encoded in the baseline instance representation. They were originally based on approximately 50 instances of relative pronouns taken from the MUC terrorism corpus, but were modified over a nine-month period to handle counter-examples as they were encountered during testing of the full information extraction system.

The results in Table 3 indicate that the baseline 1-nn system (78.8% correct) performs as well as the best default rule (77.6% correct) and below that of the hand-coded heuristics (80.5% correct). Both χ^2 and McNemar's significance tests indicate, however, that all three systems are indistinguishable from one another in terms of statistical significance. Although the accuracies of the baseline representation and the Most Recent Constituent default rule are quite close, their behavior is qualitatively different. As expected, the baseline representation performs markedly worse than the default rule for antecedents that immediately precede relative pronoun; however, it performs markedly better than the default rule for complex antecedents (conjunctions and appositives), for subject antecedents, and for detecting when "who" is not being used as a relative pronoun (antecedent = *none*). The final row in Table 3 is explained in the next section.

4.4. Comparison to IG-CBL

This section compares the performance of the baseline case-based learning algorithm to IG-CBL (Cardie & Howe, 1997), a feature weighting algorithm that has shown good performance across a number of natural language learning tasks. IG-CBL is a weighted k-nearest neighbor algorithm that is a straightforward composition of two existing approaches for feature selection and feature weighting. IG-CBL first uses a decision tree for feature selection as described in Cardie (1993) and briefly below. The goal in this step is to prune features from the representation so that the case-based learning algorithm can ignore them entirely. IG-CBL then assigns each remaining feature a weight according to its information gain across the training cases as done in the IB1-IG algorithm (Daelemans, van den Bosch, & Zavrel, 1999). The intent here is to weight each feature relative to its overall importance in the data set. There are three steps to the IG-CBL training phase:

1. *Create the case base.* For this, we simply store all of the training instances.
2. *Use the training instances to create a decision tree for the learning task.* Our experiments use C4.5 (Quinlan, 1993).
3. *Compute feature weights for use during case retrieval.* For each feature, f, we compute a weight, w_f, as follows:

$$w_f = G(f) \quad \text{if } f \text{ is in the tree of step 2;}$$
$$w_f = 0 \quad \text{otherwise;}$$

where $G(f)$ is the information gain ratio of f as computed across all training instances by C4.5.

To apply the IG-CBL algorithm to the relative pronoun data set, we first normalize the instances with respect to the entire feature set, filling in a *nil* value for missing features. After training, the class value for a novel instance is determined using a weighted k-nn case retrieval algorithm identical to that of the baseline case-based learning algorithm except that feature weights are computed as above.

We see from Table 3 that the IG-CBL feature-weighting approach works very poorly for the relative pronoun task although it has worked well for other natural language learning problems including part-of-speech tagging, semantic class tagging, prepositional phrase attachment, grapheme-to-phoneme conversion, and noun phrase chunking (Cardie, 1993; Daelemans et al., 1999). We believe that IG-CBL works poorly because of the large number of antecedent classes and because the information gain bias is not appropriate for the relative pronoun task, especially with normalized instances that contain mostly missing values. We will see better performance of IG-CBL on the data sets of Section 7.

5. Incorporating the cognitive biases

In the following subsections, we modify the baseline representation in response to three cognitive biases and measure the effects of those changes on the learning algorithm's ability to predict relative pronoun antecedents.

5.1. Incorporating the subject accessibility bias

A number of studies in psycholinguistics have noted the special importance of the first item mentioned in a sentence (e.g., Gernsbacher, Hargreaves, & Beeman, 1989; Carreiras, Gernsbacher, & Villa, 1995). In particular, it has been shown that the accessibility of the first actor of a sentence remains high even at the end of a sentence (Gernsbacher et al., 1989). The effect of this *subject accessibility bias* on processing relative clauses was also noted in King and Just (1991) and is an example of a more general *focus of attention bias*. In computer 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 propose to incorporate the subject accessibility bias into the baseline case representation by increasing the weights for any features associated with the subject of the clause preceding the relative pronoun. Weights for the subject features

Table 4. The effect of the subject accessibility bias on relative pronoun resolution (% correct).

Baseline	Subject wt = 2	Subject wt = 5	Subject wt = 8	Subject wt = 12
78.8	76.7	75.5	76.3	75.9

are increased as a function of a fixed increment, the *subject weight*. More specifically, the subject weight is divided evenly across all features associated with the subject (i.e., S, S-SEM, and possibly S-PUNC or S-MARK) and then added to the original weight for each subject feature.

Table 4 shows the effects on relative pronoun resolution when the subject weight is 2, 5, 8, and 12: incorporation of the subject accessibility bias never improves relative pronoun disambiguation when compared to the baseline representation, although dips in performance are never statistically significant using the χ^2 test. McNemar's test indicates significant differences between the baseline representation and the subject-weighted representation when the weight is 5 or 12. Higher subject weights were tested, but provide no improvement in performance.

At first these results may seem surprising, but the baseline representation produced by CIRCUS already somewhat encodes the subject accessibility bias by explicitly recognizing the subject as a major constituent of the sentence (i.e., S) rather than labeling it merely as a low-level noun phrase (i.e., NP). (Removing this bias from the baseline representation causes a drop in performance from 78.8% to 75.5%.) It may be that the original encoding of the bias is adequate or that additional modifications to the baseline representation are required before the subject accessibility bias can have a positive effect. In addition, the subject accessibility bias affects a relatively small number of features. Some cases have no subject features because no subject was identified in the clause preceding the relative pronoun. For these cases, the subject accessibility bias plays no role at all. An analysis of errors indicates that the baseline representation performs slightly better than the subject-weighted representation (weight = 2) across all of the major antecedent types.

5.2. Incorporating the recency bias

In processing language, people consistently show a bias towards the use of the most recent information (e.g., Kimball, 1973; Nicol, 1988; Gibson, 1990; Gibson et al., 1993). In particular, Frazier and Fodor (1978), Cuetos and Mitchell (1988), and others have investigated the importance of recency in finding the antecedents of relative pronouns. They found that for English there is a preference for choosing the most recent noun phrase as the antecedent in sentences where the antecedent of the relative pronoun is ambiguous. For example, in sentences like *The journalist interviewed the daughter of the colonel who had the accident*, people assume that "who" refers to "the colonel" rather than "the daughter of the colonel."

The feature selection algorithm translates this recency bias into representational changes for the training and test instances in two ways. The first is a direct modification of the feature set; the second modifies the weights to indicate a constituent's distance from the relative pronoun. In the first approach, we rename the features according to the position of

Table 5. Incorporating the recency bias using a right-to-left labeling.

Baseline representation	Sentence	Right-to-left labeling
(S *t*)	it	(S *t*)
(S-SEM *entity*)		(S-SEM *entity*)
(V *t*)	was	(V *t*)
(DO *t*)	the hardliners	(NP-2 *t*)
(DO-SEM *human*)		(NP-2 *human*)
(DO-PP-1 *t*)	in Congress	(PP-1 *t*)
(DO-PP-1-PREP *in*)		(PP-1-PREP *in*)
(DO-PP-1-SEM *entity*)		(PP-1-SEM *entity*)
(PREV-TYPE *prepositional-phrase*)	who . . .	(PREV-TYPE *prepositional-phrase*)

the associated constituent relative to the wh-word. As part of this renaming, subjects are relabeled as noun phrases (NPs) unless there is a verb in the clause that unambiguously denotes the presence of a subject. This establishes a right-to-left (r-to-l) labeling of constituents rather than the left-to-right labeling that is more natural for the parser and that is espoused in the baseline instance representation. Table 5 shows the effect of the new r-to-l labeling for one example: "It was the hardliners in Congress, who . . . " The leftmost and rightmost columns indicate the original baseline and r-to-l recency representations, respectively. The r-to-l labeling for "in Congress," for example, is via the PP-1 attributes because it is a prepositional phrase one position to the left of "who." In the original representation, that phrase is labeled with respect to the direct object that precedes it. Similarly, "the hardliners" receives the attribute NP-2 in the r-to-l recency representation because it is a noun phrase two positions before "who." The r-to-l ordering yields a different feature set and, hence, a different instance representation. Intuitively, the r-to-l representation provides a more uniform encoding of the immediate context of the relative pronoun. Consider, for example, the following sentences:

It was a message from *the hardliners* in Congress **who** . . .
It was from *the hardliners* in Congress **who** . . .

From the point of view of relative pronoun resolution, the two sentences seem very similar—"who" refers to "the hardliners" in each case. The r-to-l labeling assigns the two most recent constituents in each sentence ("in Congress" and "from the hardliners") the same attributes—PP-1 and PP-2. It also assigns the same class value to the instances for each example: the antecedent is the PP-2 constituent. The baseline epresentation, on the other hand, labels the most recent constituents and the antecedents for each instance with distinct attributes, making it less likely that one case would be retrieved in response to the other.

In the second approach to incorporating the recency bias, we increment the weight associated with each feature as a function of its proximity to the wh-word (Table 6). To create a recency-biased weight vector, the feature associated with the element closest to the relative

Table 6. Incorporating the recency weights (with a maximum weight of ten).

Phrase	Attributes	Baseline weight	Recency weight	Final weight
It	S	1	2	3
	S-SEM	1	3	4
was	V	1	4	5
the hardliners	DO	1	5	6
	DO-SEM	1	6	7
in Congress	DO-PP-1	1	7	8
	DO-PP-1-PREP	1	8	9
	DO-PP-1-SEM	1	9	10
	PREV-TYPE	1	10	11
who . . .				

Table 7. The effect of the recency bias on relative pronoun resolution (% correct). The boldface entry indicates significance with respect to the baseline representation, the hand-coded heuristics, and the recency weighting representation.

Baseline	Hand-coded heuristics	R-to-L labeling	Recency weighting	R-to-L + RecWt
78.8	80.5	81.3	80.1	**85.1**

pronoun receives a weight of ten,[2] and the weights are decreased by one for each of the preceding features until reaching zero. The recency weights are then added to the original baseline feature weights to produce the final weight vector. When using a different NLP system to generate cases, it may make more sense to implement recency weighting by assigning the same proximity-based weight to all features associated with a single phrase. This alternative does not work as well as the proposed recency weighting scheme for our NLP system and the relative pronoun data set.

The results of experiments that use each of the recency representations separately and in a combined form (R-to-L + RecWt) are shown in Table 7. To combine the two implementations of the recency bias, the system first relabels the attributes of an instance using the r-to-l labeling and then initializes the weight vector using the recency weighting procedure described above. The table shows that the recency weighting and r-to-l labeling have relatively little effect on the prediction of wh-word antecedents when applied individually: the increases in performance for these runs over the baseline are not statistically significant. The combined representation, however, improves performance significantly with respect to the baseline instance representation, the hand-coded heuristics, and the recency weighting representation (for both significance tests). McNemar's test also indicates significant differences with respect to the standalone r-to-l labeling representation.

We believe that the combined recency bias performs well because recency effects are very strong for relative pronoun resolution and because the individual implementations

of the recency bias complement one another. In particular, the representation of the local context of the wh-word provided by the r-to-l labeling is critical for finding antecedents. The recency weighting representation lacks such a representation of local context, but provides an additional emphasis on those constituents closest to the relative pronoun.

One can get a sense of the broad changes to the instance space caused by the r-to-l recency labeling by re-examining some of the original data set characteristics after applying this bias. As described in Section 4.2, the data set encoded using the baseline case representation exhibits 60 distinct antecedents. After incorporating the r-to-l recency bias, this number is reduced to 39. In addition, the number of instances with unique antecedents is similarly reduced—from ten to one. In spite of these reductions in data set complexity, the number of instance types remains about the same: of the 241 cases, 184 (76%) are unique vs. 186 (77%) using the baseline representation.

In an analysis of the results, we see that for 25 test cases, the combined recency bias is correct when the baseline representation is incorrect; the reverse is true only ten times. In general, the combined representation does markedly better than the baseline in recognizing when "who" is not being used as a relative pronoun. Although the combined representation performs better than the baseline for antecedents at all distances from the relative pronoun, over half of the differences involve 'middle distance' antecedents that are two or three phrases before the relative pronoun. Two instances that the combined recency representation gets correct when the baseline does not are listed here. (Subscripts denote the antecedent phrases selected using each representation. A subscript of *both* means that both the baseline and combined recency representations selected the phrase as a component of the antecedent.)

> Ordonez added: I was aware that Pena wanted to get rid of somebody, but $\text{I}_{baseline}$ never learned **who** they were going to kill until . . . (Correct antecedent is $\text{none}_{combined\_recency}$).

> Spaniard Jose Maria Martinez_{both}, Frenchman $\text{Roberto Lisandy}_{combined\_recency}$, and Italian $\text{Dino Rossy}_{combined\_recency}$, **who** were staying . . .

5.3. 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 (1980; 1983) show that working memory capacity affects a subject's ability to find the referents of pronouns over varying distances. Also, 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.

It should be clear that the baseline instance representation for the relative pronoun task does not make use of short term memory limitations: the learning algorithm uses all available features during case retrieval. Short term memory studies, however, do not explicitly state

Table 8. The effect of the restricted memory bias on relative pronoun resolution (% correct).

Baseline	Memory limit = 3	Memory limit = 5	Memory limit = 8	Memory limit = 12
78.8	73.4	75.3	77.3	77.8

what the short term memory limit should be—it appears to vary from five to nine depending on the cognitive task. It may also depend on the size and type of the 'chunks' that have to be remembered. In addition, the short term memory bias alone does not state which features to keep and which to discard: King and Just hypothesize the existence of interactions between short term memory limitations and attentional biases (like the subject accessibility bias) that allow certain semantic representations in a sentence to remain active over long distances.

This argues for an implementation of a *restricted memory* bias that accommodates the influence of other cognitive biases. Given a memory limit of n, the restricted memory bias selects the n features with the highest weights, choosing randomly in case of ties. It then sets the weights for the remaining features to zero. This effectively prunes all but the n selected features from the instance representation. In the baseline representation where all features have a weight of one, the restricted memory bias is as likely to discard relevant features as it is to discard irrelevant features: we expect that this bias will have a positive impact on performance only when it is combined with cognitive biases that provide additional feature relevance information. Another version of the restricted memory bias might select all features associated with n phrases rather than selecting n features. However, this alternative would be more difficult to implement in the presence of cognitive biases that modify weights for individual features rather than for entire constituents.

Table 8 shows results for the restricted memory bias with n set to one of 3, 5, 8, and 12. (Because of the random component of the restricted memory bias, results are averages across five leave-one-out cross-validation runs.) It is clear from the table that this bias degrades the ability of the system to predict relative pronoun antecedents although the drop in performance is statistically significant only when $n = 3$ using the χ^2 test. (McNemar's test shows no significant differences in performance with respect to the baseline.) These results are not surprising given the random feature selection imposed by the restricted memory bias when applied in isolation. In general, the effect of the restricted memory bias depends on the number of features in the test case and the memory limit. The average number of features per instance is 8.9; the maximum number of features in any instance is 31. Whenever the number of test case features is within the memory limit, the restricted memory bias has no effect; as the limit is increased, the performance approaches that of the baseline representation.

5.4. Discussion of results

Table 9 provides a summary of the best-performing variation of each bias implementation as well as the best baseline systems. Individually, none of the cognitive bias implementations significantly improves the accuracy of relative pronoun antecedent prediction over the

Table 9. Individual cognitive bias summary for relative pronoun resolution. The boldface entry indicates statistical significance with respect to the baseline representation and the hand-coded heuristics.

Cognitive bias or baseline system	Parameters	% Correct
Baseline systems		
Best default rule	—	77.6
Hand-coded heuristics	—	80.5
Baseline representation (no biases, 1-nn)	—	78.8
Single biases		
Subject accessibility	Subject wt = 2	76.7
Restricted memory	Memory limit = 12	77.8
Recency (r-to-l labeling)	—	81.3
Recency (recency weighting)	Max wt = 10	80.1
Combining bias implementations		
Recency (r-to-l + recency weighting)	Max wt = 10	**85.1**

baseline representation. Neither the subject accessibility nor the restriced memory biases is able to boost performance. Increases in performance from the recency weighting and r-to-l labeling when applied in isolation are not statistically significant. Only when combining both implementations of the recency bias—recency weighting and r-to-l labeling of features—do we obtain a representation that produces the first significant performance gains with respect to the baseline case representation. This combined recency representation also significantly outperforms the hand-coded heuristics. The next section presents and evaluates two automatic methods for selecting and combining all three cognitive biases. It is here that we might expect to see further increases in performance if the biases are complimentary.

6. Cognitive bias selection

The last section showed that, in spite of results on human language processing, cognitive biases are not always useful when applied in isolation. Nonetheless, we found that gains in performance for the relative pronoun resolution task can be achieved by applying more than one cognitive bias to the instance representation: e.g., when both implementations of the recency bias were instantiated simultaneously. Furthermore, the same experiments indicate that cognitive bias interactions may make it difficult to determine which biases are relevant to a particular learning task. For example, the combined recency representation performed quite well in spite of the relatively small gains produced by each recency bias in isolation. This section, therefore, presents two automated methods for cognitive bias selection—Greedy Bias Selection and Exhaustive Bias Selection—each of which makes different assumptions as to the independence of individual cognitive biases. The approaches also reflect a potential tradeoff between the quality of the selected bias combination and the computing time required to make the selection.

1. Divide the data D into n partitions. For each partition D_{test},
   (A) Let D_{test} be the test partition and $D_{train} = D - D_{test}$, the training data.
   (B) Divide D_{train} into m partitions.
      i. For each partition D_{train_s}
         A. Let D_{train_s} be the selection data and $D_{train_l} = D_{train} - D_{train_s}$, the learning data.
         B. **Apply each cognitive bias combination (and associated parameter settings) to D_{train_l} in turn. Test on D_{train_s}.**
      ii. For each bias combination tested, compute its average accuracy across the m selection data partitions.
      iii. Select the bias combination with the highest average accuracy.
   (C) Apply the selected bias(es) on D_{train}; test on D_{test}.
2. Return the average of the accuracies for each D_{test} partition.

Figure 2. The cross-validation algorithm for bias selection. The greedy and exhaustive bias selection algorithms differ only in the method each uses to instantiate the boldface step above.

6.1. The bias selection algorithms

At a high level, both the greedy and exhaustive bias selection methods employ nested cross validation as suggested in Schaffer (1993) and outlined in figure 2. The difference in the bias selection algorithms lies in the methods each uses to instantiate the highlighted step of the general algorithm. This step effectively determines the structure of the bias space and the order in which it should be searched.

In general, the greedy bias selection algorithm (figure 3) operates by incorporating the best of the remaining individual biases, one at a time, while accuracy on the selection data improves or remains constant. For the relative pronoun data set, the Biases parameter (as well as Available-Biases) includes all three cognitive biases under a variety of parameter settings, e.g., Biases = {r-to-l labeling, combined recency with max wt = 10, subject accessibility with wt = 2, . . . }. During each iteration (step 5), one of the available biases is selected for incorporation into the instance representation. After each iteration, alternative variations of the newly selected bias are removed from the set of Available-Biases (step 18). If, for example, the recency-weighting representation with a maximum weight of ten were the first bias selected, then all recency biases are deleted from consideration for the next iteration of bias selection.

The exhaustive bias selection algorithm operates simply by testing all combinations of all biases and associated parameter settings under consideration. More specifically, the exhaustive approach to bias selection instantiates the boldface step of the general bias selection algorithm (figure 2) as follows:

1. Given an initial set of cognitive biases and associated parameter settings, create a list, L, of all possible combinations thereof.

```
Greedy ( Biases, Learning-Data, Selection-Data )
 1. /* Initializations: */
 2.      Available-Biases = Biases
 3.      Sel-Biases = {}                          /* biases selected thus far */
 4.      Sel-Biases-Acc = 0                       /* accuracy of Sel-Biases */

 5. While Available-Biases
 6.      Best-Bias = nil; Best-Bias-Acc = 0

 7.      /* Find best remaining bias */
 8.      For each New-Bias in Available-Biases,
 9.           Apply Sel-Biases and New-Bias to the Learning-Data.
10.           New-Bias-Acc = accuracy of new representation on Selection-Data
11.           If New-Bias-Acc > Best-Bias-Acc
12.                Best-Bias = New-Bias; Best-Bias-Acc = New-Bias-Acc

13.      /* Exit if incorporating Best-Bias caused drop in accuracy */
14.      If Best-Bias-Acc < Sel-Biases-Acc, Return Sel-Biases and Sel-Biases-Acc

15.      /* Otherwise incorporate Best-Bias into Sel-Biases */
16.      Sel-Biases = Sel-Biases ∪ Best-Bias; Sel-Biases-Acc = Best-Bias-Acc

17.      /* Remove all variations of Best-Bias from consideration */
18.      Available-Biases = remove-variants (Available-Biases, Best-Bias)
19. Return Sel-Biases and Sel-Biases-Acc
```

Figure 3. Greedy bias selection algorithm. Instantiation of boldface step of figure 2.

2. For every bias combination in L,
   (A) Apply the bias combination to the learning data.
   (B) Test on the selection data.

In comparison to greedy bias selection, exhaustive bias selection makes few assumptions about cognitive bias interactions; on the other hand, the method requires much more computing time to select the appropriate bias combination for a particular data set. When testing n cognitive biases, each of which has m parameter settings to be considered, greedy bias selection will test $O(n^2m)$ bias/parameter setting combinations. The exhaustive approach, on the other hand, requires testing $O(m^n)$ combinations.

6.2. Merging bias representations

Both the greedy and exhaustive bias selection algorithms require an ordered procedure for merging the representations produced by two or more individual biases (step 9 in the greedy algorithm). This is accomplished as follows:

1. Incorporate any bias that relabels or adds attributes (e.g., r-to-l labeling).

2. Incorporate biases that modify feature weights by adding the weight vectors proposed by each bias (e.g., recency weighting, subject accessibility).
3. Incorporate biases that discard features (e.g., restricted memory bias).

As was the case with representations that included one cognitive bias, the combined bias representations are created automatically. The user specifies only the list of biases to be applied to the problem and any associated parameters.

6.3. Results

Table 10 shows the results of applying the bias selection algorithms on the relative pronoun disambiguation task. For these runs, both levels of cross validation used ten partitions ($n = 10$, $m = 10$). For each algorithm, the table lists the selected biases and the number of partitions for which the combination was selected. The biases considered are shown at the bottom of the table. The greedy algorithm makes a fairly uniform selection of cognitive biases across partitions. Not surprisingly, the combined recency bias is always selected first. The restricted memory (RM) bias was always selected next with memory limits ranging from five to seven. Finally, the subject accessibility bias was selected for four out of ten

Table 10. Bias selection results for the relative pronoun resolution task. For each bias selection algorithm, the table shows the selected biases and the number of 10-fold cross validation partitions for which the bias was selected. Accuracy refers to average % correct across the ten partitions.

Selected biases			Number of partitions
Greedy selection			
Rec Combo (max wt = 10)	RM (limit = 7)		5
Rec Combo (max wt = 10)	RM (limit = 5)	Subj (wt = 3)	2
Rec Combo (max wt = 10)	RM (limit = 6)		1
Rec Combo (max wt = 10)	RM (limit = 6)	Subj (wt = 3)	1
Rec Combo (max wt = 10)	RM (limit = 7)	Subj (wt = 3)	1
			Accuracy: 89.2 ± 5.5
Exhaustive selection			
Rec Combo (max-wt = 10)	RM (limit = 5)	Subj (wt = 2)	7
Rec Combo (max-wt = 10)	RM (limit = 8)	Subj (wt = 2)	2
Rec Combo (max-wt = 10)	RM (limit = 8)	Subj (wt = 4)	1
			Accuracy: 89.6 ± 5.9
Biases tested:			
Right-to-left recency labeling (R-to-L)			
Recency weighting (RecWt): max wt = {10, 20}			
Combined R-to-L and RecWt (Rec Combo): max wt = {10, 20}			
Subject accessibility (Subj): wt = {2, 3, 4, 6, 7, 10, 12}			
Restricted memory (RM): limit = {3, 4, 5, 6, 7, 8, 10, 12, 15, 20, 25, 30}			

partitions with a subject weight of three in all cases. The average accuracy of the representations created using greedy bias selection is 89.2%, which significantly outperforms the baseline case representation, the best default rule, the hand-coded heuristics, and all of the individual cognitive biases. It also significantly outperforms the combined recency biases.[3]

Exhaustive bias selection also chooses a fairly stable set of cognitive biases across partitions: it selects the combined recency representation with a maximum weight of ten; the restricted memory bias with a relatively small memory limit (usually five); and the subject accessibility bias with a small subject weight (usually two). The average accuracy of the representations created using exhaustive bias selection is 89.6%. Like greedy bias selection, exhaustive bias selection performs significantly better than the baseline case representation, the best default rule, all of the individual biases, the hand-coded heuristics, and the combined recency biases. Significance tests indicate no difference in performance when compared to the greedy selection algorithm. The running time for one (outer-level) partition of exhaustive bias selection with $n = 10$ and $m = 10$ is about 17 minutes on an Ultra Sparc 5. In contrast, one partition of greedy bias selection takes about two minutes.

6.4. Discussion and summary of results

Table 11 summarizes the results of applying the cognitive bias approach to feature set selection for the relative pronoun task. The table clearly shows that performance of the learning algorithm increases steadily as relevant biases are added to the baseline representation. As noted earlier, we see mild, but statistically insignificant gains when the r-to-l recency bias is applied in isolation. When merged with recency weighting, however, there are significant gains in performance with respect to the baseline representation. Incorporation of the subject accessibility and restricted memory biases using either greedy or exhaustive bias selection provides additional improvements in performance. The feature set selection method obtains improvements in spite of the fairly small data set size. With a larger data set, we would expect better representation of, and hence better performance on, infrequently occurring antecedent types.

In general, the all-biases representation performs markedly better than the combined recency representation whenever the clause that precedes the relative pronoun is fairly long. The average number of features for these cases is 14.5; in contrast, the average

Table 11. Bias selection summary for relative pronoun resolution. Results in boldface indicate significant increases with respect to the baseline representation. The * indicates significant improvements with respect to the combined recency representation.

No biases	Baseline representation	78.8
Best individual bias	Recency (r-to-l)	81.3
	Combined recency (r-to-l + recency weighting)	**85.1**
Combining two or more biases	Greedy bias selection	**89.2***
	Exhaustive bias selection	**89.6***

number features per case for the entire data set is 8.9. It is likely that the restricted memory bias is responsible for most of this improvement: it tends to prune features for distant constituents in the all-biases representation, allowing the case-based learning algorithm to concentrate on recent phrases during case retrieval. Furthermore, 73% of the improvement between the combined recency and all-biases representations is due to better performance on syntactically ambiguous relative pronoun constructs (see Section 4.2). Two such examples follow:

> The government publicly shows the horror$_{recency}$ of women$_{all-biases}$ **who** have been raped in the prisons . . .
>
> They also recommend that the persons who are going to carry out the abductions should select the victims from among politicians and members$_{all-biases}$ of the Colombian bourgeoisie$_{recency}$ **who** have never distinguished themselves . . .

The two example shows that these syntactically ambiguous cases can be semantically difficult as well: it is often hard for a person to provide consistent antecedent information in the presence of collective or mass nouns (e.g., group, members). In addition, it is sometimes necessary to read the relative clause in order to disambiguate the relative pronoun. Our current case representation, however, includes no features for phrases in the relative clause itself, making it difficult to handle this type of ambiguity.

This section also showed that both the greedy and exhaustive search in conjunction with cross validation can be used for automatic bias selection. In particular, the experiments indicate that greedy bias selection may be adequate whenever interactions among cognitive biases are sufficiently limited. Finally, while exhaustive bias selection performs slightly better, the small gains in performance over greedy selection may not be worth the increase in running time.

7. Additional data sets

Thus far, we have concentrated on evaluating the cognitive bias approach to feature set selection using a single data set and three cognitive biases. In this section, we show that the approach is effective for tasks other than relative pronoun resolution. In particular, we apply the approach to three additional language learning tasks and make two more cognitive biases available to the learning algorithm.

7.1. Handling unknown words

The additional data sets correspond to three lexical tagging tasks that address the problems encountered by a natural language processing system when it reaches unknown words, i.e., words not in the system lexicon. Given the context in which each unknown word occurs, our NLP system must predict the word's part of speech as well as its general and specific semantic class. Assume, for example, that the word "general" was an unknown word and that the NLP system encountered the following two sentences from the MUC-3 information

extraction corpus:

The *general* concern was that children might be killed.
The terrorists killed *General* Bustillo.

In the first sentence, the system should indicate that the part of speech of "general" is an adjective; in the second sentence, it is a noun modifier. Similarly, given a two-level semantic feature hierarchy, the system should determine that "general" is being used in its "universal entity" sense in the first case, but as a "person:military officer" in the second.

Each lexical tagging data set contains 2056 cases. Like the relative pronoun data set, instances are created automatically by the CIRCUS parser. Each represents the context in which the system encounters the unknown word in its left-to-right traversal of 120 randomly selected sentences from the MUC business joint ventures corpus (MUC-5, 1994). Each case has 34 features. In the example of figure 4, only the non-nil features are shown. Twenty-one features describe the local context in which the test word occurred: these are the morphology (MORPHOL) of the unknown word and the part of speech (P-O-S), semantic classes (GEN-SEM, SPEC-SEM), information extraction concept (IE-CONCEPT), and actual lexical item (WORD) corresponding to the words in a five-word window centered on the unknown word. In our data sets, there are 18 possible parts of speech and 11 domain-specific concept types. GEN-SEM and SPEC-SEM correspond to entries in a two-level semantic class hierarchy defined for use in the joint ventures domain. The hierarchies in our experiments have 14 general semantic features and 42 specific semantic features. The remaining 13 features of the lexical tagging cases encode the semantic features and information extraction concepts for the major

Phrases	Features	
Toyota Motor Corp.	(SUBJECT-GEN-SEM	*joint-venture-entity*)
	(SUBJECT-SPEC-SEM	*company-name*)
has set up	(LAST-PHRASE-SYN-TYPE	*verb*)
a	(PREV2-WORD	*a*)
	(PREV2-P-O-S	*determiner*)
joint	(PREV1-WORD	*joint*)
	(PREV1-P-O-S	*adjective*)
	(PREV1-GEN-SEM	*entity*)
venture		
firm	(FOL1-WORD	*firm*)
	(FOL1-P-O-S	*noun*)
	(FOL1-GEN-SEM	*joint-venture-entity*)
with	(FOL2-WORD	*with*)
	(FOL2-P-O-S	*preposition*)
Class values:		
part-of-speech	NOUN-MODIFIER	
gen-sem	ENTITY	
spec-sem	NIL	

Figure 4. Baseline instance representation for the lexical tagging case for "*venture*" in "*Toyota Motor Corp. has set up a joint venture firm with Yokogawa Electric Corp. . . .*" Only the non-nil entries in the representaton are shown.

syntactic constituents (i.e., the subject, verb, direct object, and most recent phrase) that have been recognized at the time that the unknown word is encountered. Finally, each case includes the three class values to be predicted—the unknown word's part-of-speech, and general and specific semantic features. As was the case for relative pronoun resolution, the case representation reflects the syntactic and semantic information available to CIRCUS as it processes a text. In this specification for handling unknown words, we treat each prediction task independently.

In general, the features for the lexical tagging tasks are very similar to those used for relative pronoun resolution. There are two main differences. First, we have encoded a richer description for the individual lexical items in close proximity to the unknown word. For the relative pronoun task, we concentrated on constituent-level representations. This difference is reasonable since the current task is a lexical task rather than a structural attachment decision. However, since two-thirds of the features in the lexical tagging data sets now represent neighboring tokens, the recency bias may have little effect. Second, we have already discarded many irrelevant features from the representation. For example, the NLP system could easily have included features for every low-level phrase it recognizes (as we did for relative pronoun resolution) and then relied on the learning algorithm to discard all irrelevant features. This preprocessing step inflates the performance of the baseline representation. In addition, there may be less of a need for biases that discard features, like the restricted memory bias. The data sets, therefore, may respond less readily to a number of the cognitive biases. This will be a good test for the bias selection algorithms, which may have to recognize that not all available biases are relevant to the problems at hand.

7.2. The semantic priming and syntactic biases

To show that our approach can support a variety of cognitive biases, we define two additional biases for use with the lexical tagging tasks. The first is *semantic priming*. Semantic priming is a well-known cognitive effect—during on-line information processing, people tend to respond more quickly to words that are semantically related to entities currently involved in the interpretation process. Our system implements semantic priming by increasing the weights for all semantic features in the baseline representation (e.g., the general or specific semantic classes of words or constituents) by some specified value. This is only a very coarse implementation of this bias, however. It encourages the case retrieval algorithm to match on semantic features, but ignores the problem of determining which entities are most pertinent at the current point in processing. For this, we will rely on the other cognitive biases. Analogously, we define a *syntactic priming* bias, which increases the weights for all features associated primarily with syntactic issues (e.g., the part of speech of words or syntactic category of constituents) by some specified value.

7.3. Results on the lexical tagging tasks

In the experiments below, we investigate the use of all five biases—recency weighting, subject accessibility, restricted memory, semantic priming, and syntactic priming—on the

lexical tagging tasks. The right-to-left labeling bias is not applicable to these data sets, since all features are already effectively labeled with respect to the unknown word. To apply the recency weighting bias, we assume that features associated with the words in the five-word window are more recent than any constituent features. Furthermore, we assume the following recency 'ranking' among terms in the five-word window: (1) unknown word features (MORPHOL); (2) features for the token immediately preceding the unknown word (PREV1); (3) features for the token immediately following the unknown word (FOL1); (4) features for the second token that precedes the unknown word (PREV2); and (5) features for the second token that follows the unknown word (FOL2). Allowing "following" tokens to incur weights from the recency bias reflects the fact that lexical decision tasks often have an associated time delay of about 200 ms (Swinney, 1979), during which time subsequent tokens can begin to be processed. All experiments employ 10-fold cross validation.[4]

7.3.1. Effect of individual biases. The results for the lexical tagging tasks using each of the cognitive biases in isolation are summarized in Table 12. For each bias, the table shows results for only the best parameter setting for that bias. In addition, the table shows the performance of two default heuristics and the IG-CBL feature-weighting algorithm as well as the baseline case representation that includes no cognitive biases. The first default

Table 12. Incorporating individual cognitive biases for the lexical tagging tasks (% correct). The boldface entries indicate significant increases with respect to the corresponding baseline representation. Significant decreases in performance are not shown.

Cognitive bias or baseline system tested	Part of speech (p-o-s)	General semantic class (gen-sem)	Specific semantic class (spec-sem)
Most frequent tag	81.5	25.6	58.1
Weighted random selection	34.3	17.0	37.3
IG-CBL	**90.3** $\pm$ 3.3	64.7 $\pm$ 5.7	73.4 $\pm$ 3.2
Baseline representation	89.0 $\pm$ 3.7	63.9 $\pm$ 5.4	74.8 $\pm$ 5.3
Recency weighting	**92.9** $\pm$ 3.4 (max = 11)	**70.4** $\pm$ 3.8 (max = 11)	**77.5** $\pm$ 4.0 (max = 11)
Semantic priming	85.7 $\pm$ 4.2 (wt = 2)	62.0 $\pm$ 5.3 (wt = 2)	74.5 $\pm$ 5.1 (wt = 2)
Syntactic priming	**90.7** $\pm$ 4.1 (wt = 6)	61.8 $\pm$ 4.4 (wt = 2)	73.0 $\pm$ 4.9 (wt = 2)
Subject accessibility	88.4 $\pm$ 3.6 (wt = 2)	63.9 $\pm$ 5.0 (wt = 2)	74.2 $\pm$ 5.1 (wt = 2)
Restricted memory	87.5 $\pm$ 3.0 (limit = 25)	61.5 $\pm$ 5.5 (limit = 25)	72.8 $\pm$ 5.1 (limit = 25)
Biases tested:			
Recency weighting: max wt = {6, 11, 25}			
Semantic priming: wt = {2, 4, 6}			
Syntactic priming: wt = {2, 4, 6}			
Subject accessibility: wt = {2, 6, 12}			
Restricted memory: limit = {6, 11, 25}			

heuristic selects the most frequently occurring class value; the second heuristic performs a weighted random selection based on class frequency. We see first that, unlike the relative pronoun data set, the baseline representation performs significantly better than the default heuristics. The IG-CBL feature-weighting approach also works well although it significantly outperforms the baseline representation only for part-of-speech tagging.

Like relative pronoun resolution, however, the recency bias provides significant increases in performance over the baseline system. This is the case for all three data sets and in spite of the fact that the baseline representation already focuses somewhat on recent items. The only other cognitive bias that boosts performance is the syntactic bias, which helps part-of-speech prediction. All other biases either significantly decrease performance on all data sets or have no effect on the lexical tagging task.

In general, the individual bias results for the lexical tagging tasks are not all that surprising. From a linguistic point of view, the recency weighting corresponds to giving preferential status to features of those lexical items that are closest to the unknown word. This is consistent with many successful lexical tagging approaches that classify tokens based only on information associated with one or two of the immediately preceding tokens.

7.3.2. Combining cognitive biases. Table 13 shows the results for combining more than one bias for each lexical tagging task using the exhaustive and greedy bias selection algorithms. Running the full nested 10-fold cross validation for exhaustive bias selection on these larger data sets (2056 vs. 241 instances) and for a fairly large number of biases and parameter settings was not feasible. Rather than limit the number of biases and parameter settings considered, however, we chose to limit m, the number of partitions used in the inner cross validation of figure 2. The running time for one outer and one inner partition (with $n = 10$; $m = 10$) is close to 11 hours for exhaustive selection and under 20 minutes for greedy selection. All results in Table 13 were obtained with ten outer partitions and two inner partitions. The results for any bias combination that includes a random component (i.e., the restricted memory bias) are averages over five such runs.

The table shows that both bias selection algorithms provide significant increases in performance with respect to the baseline representation. The combined biases, however, never significantly improve performance over the recency bias. Table 14 shows why: for the most part, the greedy bias selection algorithm focuses on recency. Recency is often the only bias selected for the lexical tagging tasks. This is the case in seven out of ten partitions for both

Table 13. Combining cognitive biases for the lexical tagging tasks (% correct). The boldface entries indicate significant increases with respect to the corresponding baseline representation.

Task	Baseline results	Greedy selection	Exhaustive selection
part-of-speech	89.0 ± 3.7	**92.9** ± 3.5	**93.0** ± 2.3
gen-sem	63.9 ± 5.4	**69.1** ± 3.6	**69.9** ± 3.8
spec-sem	74.8 ± 5.3	**78.1** ± 3.8	**78.5** ± 4.3

Table 14. Summary of greedy bias selection results for lexical tagging. For each learning task, we show the selected biases and the number of cross-validation partitions for which the corresponding bias combination was selected.

Natural language task	Selected biases	Number of partitions
p-o-s	Bias 1: recency weighting	10/10
	with max wt = 11	7/10
	Bias 2: restricted memory	8/10
	with limit between 8 and 21	8/10
gen-sem	Bias 1: recency weighting	10/10
	with max wt = 11	9/10
	Bias 2: none	7/10
spec-sem	Bias 1: recency weighting	10/10
	with max wt = 35	6/10
	Bias 2: none	7/10

Biases tested:

Recency weighting: max wt = {6, 11, 18, 25, 35}

Semantic priming: wt = {2, 4, 6}

Syntactic priming: wt = {2, 4, 6}

Subject accessibility: wt = {2, 3, 6, 8, 14, 21}

Restricted memory: limit = {3, 6, 8, 11, 14, 21, 25}

semantic tagging tasks and suggests that the features associated with neighboring words are critical for the lexical tagging tasks. The restricted memory bias is also often selected for part-of-speech tagging, but not for the semantic tagging tasks. The effect of this bias is to discard features associated with the major constituents in the sentence. In general, these are semantically-based features that may be more useful for the semantic tagging tasks. We also see that part-of-speech and general semantic tagging prefer a smaller context. The more detailed predictions required by specific semantic tagging, however, require extending the recency weighting across all context features, i.e., increasing the maximum weight.

Analysis of the exhaustive bias selection results (no table) shows similar trends: (1) the recency bias plays the most prominent role; (2) the selected recency weights focus on a three-word window for part-of-speech and general semantic class tagging; (3) the selected recency weights affect the entire context for specific semantic class tagging; (4) the restricted memory bias is selected for part-of-speech tagging, but with a slightly higher memory limit than that chosen by the greedy algorithm. There is, however, one major difference between the bias selection algorithms: the exhaustive approach tends to include more biases than the greedy approach.

Overall, our results appear to indicate that, given CIRCUS's linguistic knowledge sources and bias implementations, the recency bias is the most important bias for lexical tagging tasks. In addition, we find that the greedy bias selection algorithm is able to select biases as well as the much more expensive exhaustive bias selection algorithm.

8. Related work

Much previous work has addressed the role of biases in machine learning algorithms. In particular, there has been recent interest in automating methods for evaluating and selecting such biases. In their overview article to a special issue of this journal on the topic, Gordon and desJardins (1995) view bias selection as searching a space of learning biases. Within their framework, the work proposed here uses cognitive processing limitations as a type of prior knowledge that guides the selection of an appropriate *representational bias* for the learning algorithm—the cognitive biases specify the set of primitive terms, or features, that define the space of allowable inductive hypotheses. Furthermore, our approach to bias selection uses greedy and exhaustive search in conjunction with cross validation as *procedural* meta-biases that order search in the representational bias space. In related work, Provost and Buchanan (1995) specify three techniques for building *inductive policies*, i.e., policies for building strategies for bias selection. Our cognitive bias approach to feature set selection makes use of all three techniques: (1) cognitive biases add structure to the bias space, e.g., the restricted memory bias limits the number of features considered by the learning algorithm; (2) the cognitive bias approach guides bias-space search, e.g., the greedy search algorithm incrementally combines the available biases; and (3) the cognitive bias approach to feature set selection constructs a learned theory across multiple biases, i.e., both bias selection algorithms combine the representations produced from individual biases according to the bias merging procedure specified in Section 6. In addition, the work presented here is innovative in the source of inspiration for the types of biases that we consider, namely cognitive preferences.

Previous work in feature set selection has relied on greedy search algorithms (e.g., Xu et al., 1989; Caruana & Freitag, 1994; John et al., 1994; Skalak, 1994) and cross validation (e.g., Maron & Moore, 1997). Our work differs from these approaches in that search occurs not in the feature space, but in the much smaller space of available cognitive biases. Search and cross validation are not used to directly select relevant subsets of features. They are used instead to select cognitive biases, which are, in turn, responsible for directing both feature selection and feature weighting.

Because the cognitive bias approach to feature weighting differs in spirit from most general methods for feature weighting (e.g., Winnow (Littlestone, 1988)), we focus here on comparisons to feature-weighting methods for case-based learning algorithms. Even within the case-based learning paradigm, however, we know of no previous work that makes use of cognitive biases to guide feature selection. In general, feature-weighting algorithms for instance-based approaches can be visualized on a continuum, from global methods that compute a single weight vector for all cases to extremely local methods that compute a different weight vector for each pair of training and test cases. A number of local weighting schemes where feature weights can vary from instance to instance (or feature value to feature value) have been proposed. The value difference metric of Stanfill and Waltz (1986) was an early machine learning algorithm that assigned a different weight to each value of a feature. In other work, Aha and Goldstone (1992) associate a different weight vector with every training case by combining globally and locally computed feature weights. The greater the similarity of a test case to the training case, the greater the emphasis of the

training case weights over the global weights. Yet another case-based learning algorithm that allows feature relevance to vary across the training instances is the RC algorithm of Domingos (1997). This algorithm uses a context-sensitive clustering method to perform feature selection rather than to assign continuous feature weights.

Still other case-based learning systems implement coarsely local feature weighting schemes that allow weights to vary at the class level. Our class distribution weighting method is one example (Howe & Cardie, 1997). It computes a different weight vector for each class in the set of training cases using statistical properties of that subset of the data. Creecy et al. (1992) use per-category feature importance to assign high weight to features that are highly correlated with the class. The IB4 (Aha, 1992) classifier also calculates a different weight vector for each class. It attempts to learn feature weights by cycling through the training instances and adjusting their values. Weights are strengthened if feature values match for instances of the same class, and weakened if the values match but the instances are of different classes.

Other methods compute query-specific weights (Wettschereck et al., 1997) by producing a different similarity metric for each test case. In recent work, we investigated the use of test-case-specific feature weights based on information gain for improving the performance of minority class instances (Cardie & Howe, 1997). Hastie and Tibshirani (1994) and Friedman (1994) also compute test-case-specific metrics; their metrics rely on discriminant analysis and recursive partitioning, respectively. In addition, Atkeson et al. (1997) create a different similarity metric for each test case, but do so for regression rather than discrete-valued classification. The cognitive bias approach to feature selection and feature weighting creates either a query-specific similarity metric or a global similarity metric depending on the data set. For the relative pronoun task, the method computes a similarity measure based on the test case: the selected biases are applied to the test case and perform feature selection and feature weighting. At first it may appear that the same is true for the lexical tagging tasks. For these data sets, however, all cases have the same number and type of attributes. The result is that the similarity metric derived by the cognitive bias approach is global in that the same features are selected and the same weights used for every case retrieval during testing.

Finally, others have examined local similarity metrics based upon domain-specific knowledge (Cain, Pazzani, & Silverstein, 1991; Skalak, 1992). In contrast, our approach uses domain-independent background knowledge in the form of general cognitive processing limitations to guide feature set selection and feature weighting. Fisher's (1987) COBWEB also has some similarities to the methods introduced here. COBWEB is a conceptual clustering system that uses a psychologically motivated, test-case-specific similarity metric to guide concept formation. In particular, it creates a polythetic classification tree with training cases at the terminal nodes, and bases most predictions on these stored cases. Because the system uses probabilistic weights to sort test cases, it effectively assigns different weights to each path in its tree and, like our approach, can apply a different similarity metric to each test case.

Case-based learning approaches to language learning (see Daelemans, 1999) also sometimes make use of one or more cognitive biases. At a minimum, an appropriate context window must be selected for inclusion in the case representation. In general, however, any

use of cognitive biases in these systems is only implicit. In addition, previous case-based learning efforts for NLP require the design of a new feature set for each natural language learning problem. Our approach marks a first step in allowing an NLP system to use the same underlying instance representation across many linguistic knowledge acquisition tasks. For a more detailed discussion of related work in NLP, see Cardie (in press).

9. Summary

The research presented here has shown that cognitive processing limitations can serve as a domain-independent source of bias to guide feature set selection for case-based learners. We have concentrated on a collection of learning tasks from natural language processing and explored the effects of five well-known cognitive biases on these tasks: (1) subject accessibility, (2) recency, (3) short term memory limitations, (4) semantic priming, and (5) syntactic priming. We have shown that cognitive biases can be automatically and explicitly incorporated into the instance representations for each natural language learning task. We have also introduced two algorithms for cognitive bias selection. The algorithms combine cross validation with greedy and exhaustive search and are able to select one or more relevant biases under appropriate parameter settings for each of the data sets tested. The representation that uses the selected bias or biases significantly outperforms the baseline representation as well as the default heuristics for each data set. In addition, it performs significantly better than an information gain-based feature set selection method known to perform well on a variety of natural language learning data sets.

Moreover, our experiments using a nearest-neighbor learning algorithm to determine the antecedents of relative pronouns indicate that the learning algorithm improves as more relevant cognitive biases are incorporated into the instance representation. For the lexical tagging tasks, the selected bias(es) were able to improve upon the baseline in spite of the fact that a number of irrelevant features had already been discarded from the baseline representation in a preprocessing step. Finally, we found that greedy bias selection worked quite well for our data sets in spite of the algorithm's limited ability to handle bias interactions. This provides evidence that sets of compatible cognitive biases can sometimes be selected for a particular learning task without enumerating and evaluating all possible cognitive bias combinations.

Still, there are additional issues that need to be addressed in future work. First, the feature set selection method should be tested on larger data sets. With larger lexical tagging data sets, for example, it should be possible to reduce the variance in performance between cross-validation partitions and possibly to see better performance of both the individual biases and the bias selection algorithms. Another important line of investigation is the development of methods that can identify cognitive bias interactions a priori so that only appropriate bias combinations need to be tested for effectiveness. Furthermore, we have tested the approach only on lexical and structural tasks in the natural language processing domain. We would expect the cognitive bias approach to feature set selection to work well for other learning tasks where the application of cognitive biases and preferences makes sense. Examples might include speech understanding (where some of the same biases

investigated here should apply), image classification (where focus of attention, brightness, and color biases could be used), and the design of adaptable user-interfaces (where focus of attention, restricted memory limitations, and recency and color preferences may play a role). We hope to apply the methods presented here to learning problems from these domains. Future work might also investigate the use of cognitive biases to aid feature weighting for algorithms other than case-based learning. Finally, it may also be possible to create an unsupervised counterpart to the inductive learning algorithm presented here. This would eliminate the need for expensive and time-consuming linguistic annotation of training texts with supervisory information.

Acknowledgments

We thank David Skalak for many helpful discussions and comments on an earlier draft of this paper. In addition, we thank the anonymous reviewers and editor David Aha for their constructive comments and suggestions. The research reported here was supported in part by NSF grant IRI–9624639.

Notes

1. There are 11 such semantic features: human, proper-name, location, entity, physical-target, organization, weapon, attack, pronoun, time, and quantity. A subset of these features are specific to the domain from which the training instances were extracted. A different set would be required for texts from a different domain.
2. The weight was chosen based on preliminary testing. Other values will be tested in subsequent sections.
3. Here we are comparing accuracies for leave-one-out vs. 10-fold cross validation, which violates training set size assumptions for the significance tests. The same significant differences were obtained, however, using 10-fold cross validation results for the baseline and single bias experiments.
4. The value of k (k-nn) for each NLP task was chosen via cross validation from among values of 1, 3, 5, 10, and 15: part-of-speech prediction, $k = 1$; general semantic class, $k = 5$; specific semantic class, $k = 10$.

References

Aha, D. W. (1992). Tolerating noisy, irrelevant, and novel attributes in instance-based learning algorithms. *International Journal of Man-Machine Studies*, *36*, 267–287.

Aha, D. W. & Goldstone, R. L. (1992). Concept learning and flexible weighting. In *Proceedings of the Fourteenth Annual Conference of the Cognitive Science Society* (pp. 534–539). Hillsdale, NJ: Lawrence Erlbaum Associates.

Aha, D. W., Kibler, D., & Albert, M. (1991). Instance-based learning algorithms. *Machine Learning*, *6*(1), 37–66.

Atkeson, C. G., Moore, A. W., & Schaal, S. (1997). Locally weighted learning. *Artificial Intelligence Review*, *11*, 11–73.

Broadbent, D. E. (1958). *Perception and communication*. London: Pergamon.

Cain, T., Pazzani, M., & Silverstein, G. (1991). Using domain knowledge to influence similarity judgement. In *Proceedings of the Case-Based Reasoning Workshop* (pp. 191–199). San Francisco, CA: Morgan Kaufmann.

Cardie, C. (1993). Using decision trees to improve case-based learning. In *Proceedings of the Tenth International Conference on Machine Learning* (pp. 25–32). San Francisco, CA: Morgan Kaufmann.

Cardie, C. (1997). Empirical methods in information extraction. *AI Magazine*, *18*(4), 65–79.

Cardie, C. (1999). Integrating case-based learning and cognitive biases for machine learning of natural language. *Journal of Experimental and Theoretical Artificial Intelligence*, *11*, 297–337

Cardie, C. & Howe, N. (1997). Improving minority class prediction using case-specific feature weights. In *Proceedings of the Fourteenth International Conference on Machine Learning* (pp. 57–65). San Francisco, CA: Morgan Kaufmann.

Carreiras, M., Gernsbacher, M. A., & Villa, V. (1995). The advantage of first mention in Spanish. *Psychonomic Bulletin and Review, 2*, 124–129.

Caruana, R. & Freitag, D. (1994). Greedy attribute selection. In *Proceedings of the Eleventh International Conference on Machine Learning* (pp. 28–36). San Francisco, CA: Morgan Kaufmann.

Cover, T. & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory, 13*, 21–27.

Creecy, R., Masand, B., Smith, S., & Waltz, D. (1992). Trading mips and memory for knowledge engineering. *Communications of the ACM, 35*, 48–64.

Cuetos, F. & Mitchell, D. C. (1988). Cross-linguistic differences in parsing: Restrictions on the use of the late closure strategy in Spanish. *Cognition, 30*(1), 73–105.

Daelemans, W. (Ed.). (1999). Special issue on case-based learning of natural language. *Journal of Experimental and Theoretical Artificial Intelligence, 11*.

Daelemans, W., van den Bosch, A., & Zavrel, J. (1999). Forgetting exceptions is harmful in language learning. *Machine Learning, 34*(1–3), 11–43.

Daneman, M. & Carpenter, P. A. (1980). Individual differences in working memory and reading. *Journal of Verbal Learning and Verbal Behavior, 19*, 450–466.

Daneman, M. & Carpenter, P. A. (1983). Individual differences in integrating information between and within sentences. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 9*, 561–584.

Domingos, P. (1997). Context-sensitive feature selection for lazy learners. *Artificial Intelligence Review, 11*, 227–253.

Elman, J. (1990). Finding structure in time. *Cognitive Science, 14*, 179–211.

Fisher, D. (1987). Knowledge acquisition via incremental conceptual clustering. *Machine Learning, 2*, 139–172.

Frazier, L. & Fodor, J. D. (1978). The sausage machine: A new two-stage parsing model. *Cognition, 6*, 291–325.

Friedman, J. H. (1994). Flexible metric nearest neighbor classification. Unpublished manuscript available at playfair.stanford.edu via /pub/friedman/README.

Gernsbacher, M. A., Hargreaves, D. J., & Beeman, M. (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, E. (1990). Recency preferences and garden-path effects. In *Proceedings of the Twelfth Annual Conference of the Cognitive Science Society* (pp. 72–79). Hillsdale, NJ: Lawrence Erlbaum Associates.

Gibson, E., Pearlmutter, N., Canseco-Gonzalez, E., & Hickok, G. (1993). Cross-linguistic attachment preferences: Evidence from English and Spanish. In *Sixth Annual CUNY Sentence Processing Conference.* Amherst, MA: University of Massachusetts.

Gordon, D. & desJardins, M. (1995). Evaluation and selection of biases in machine learning. *Machine Learning, 20*(1–2), 5–22.

Hastie, T. J. & Tibshirani, R. J. (1994). Discriminant adaptive nearest neighbor classification. Unpublished manuscript available at playfair.stanford.edu as /pub/hastie/dann.ps.Z.

Howe, N. & Cardie, C. (1997). Examining locally varying weights for nearest neighbor algorithms. In *Proceedings of the Second International Conference on Case-Based Reasoning* (pp. 455–466). Berlin: Springer.

John, G., Kohavi, R., & Pfleger, K. (1994). Irrelevant features and the subset selection problem. In *Proceedings of the Eleventh International Conference on Machine Learning* (pp. 121–129). San Francisco, CA: Morgan Kaufmann.

Kimball, J. (1973). Seven principles of surface structure parsing in natural language. *Cognition, 2*, 15–47.

King, J. & Just, M. A. (1991). Individual differences in syntactic processing: The role of working memory. *Journal of Memory and Language, 30*, 580–602.

Langley, P. (1996). *Elements of machine learning*. San Francisco, CA: Morgan Kaufmann.

Lehnert, W. (1990). Symbolic/subsymbolic sentence analysis: Exploiting the best of two worlds. In J. Barnden & J. Pollack (Eds.), *Advances in connectionist and neural computation theory*. Norwood, NJ: Ablex Publishers.

Lehnert, W., Cardie, C., Fisher, D., Riloff, E., & Williams, R. (1991). University of Massachusetts: Description of the CIRCUS system as used in MUC-3. In *Proceedings of the Third Message Understanding Conference (MUC-3)* (pp. 223–233). San Mateo, CA: Morgan Kaufmann.

Littlestone, N. (1988). Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, *2*, 285–318.

Maron, O. & Moore, A. (1997). The racing algorithm: Model selection for lazy learners. *Artificial Intelligence Review*, *11*(1–5), 193–225.

Miller, G. A. (1956). The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, *63*(1), 81–97.

Mitchell, T. (1997). *Machine learning*. New York: McGraw-Hill.

MUC-3 (1991). *Proceedings of the Third Message Understanding Conference (MUC-3)*. San Mateo, CA: Morgan Kaufmann.

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

Newport, E. (1990). Maturational constraints on language learning. *Cognitive Science*, *14*, 11–28.

Nicol, J. (1988). Coreference processing during sentence comprehension. Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, MA.

Provost, F. & Buchanan, B. (1995). Inductive policy: The pragmatics of bias selection. *Machine Learning*, *20*(1–2), 35–62.

Quinlan, J. R. (1993). *C4.5: Programs for machine learning*. San Mateo, CA: Morgan Kaufmann.

Schaffer, C. (1993). Overfitting avoidance as bias. *Machine Learning*, *10*(2), 153–178.

Skalak, D. (1992). Representing cases as knowledge sources that apply local similarity metrics. In *Proceedings of the Fourteenth Annual Conference of the Cognitive Science Society* (pp. 325–330). Hillsdale, NJ: Lawrence Erlbaum Associates.

Skalak, D. (1994). Prototype and feature selection by sampling and random mutation hill climbing algorithms. In *Proceedings of the Eleventh International Conference on Machine Learning* (pp. 293– 301). San Francisco, CA: Morgan Kaufmann.

Stanfill, C. & Waltz, D. (1986). Toward memory-based reasoning. *Communications of the ACM*, *29*, 1213–1228.

Swinney, D. A. (1979). Lexical access during sentence comprehension: (Re)consideration of context effects. *Journal of Verbal Learning and Verbal Behavior*, *18*, 645–660.

Wettschereck, D., Aha, D. W., & Mohri, T. (1997). A review and empirical evaluation of feature-weighting methods for a class of lazy learning algorithms. *Artificial Intelligence Review*, *11*, 273–314.

Wilson, R. A. & Keil, F. (Eds.). (1999). *The MIT encyclopedia of the cognitive sciences*. Cambridge, MA: MIT Press.

Xu, L., Yan, P., & Chang, T. (1989). Best first strategy for feature selection. In *Ninth International Conference on Pattern Recognition* (pp. 706–708). IEEE Computer Society Press.

Received September 2, 1998
Revised October 19, 1998
Final manuscript August 6, 1999