Feature Selection for Case-Based Learning: A Cognitive Bias Approach

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

Abstract

Experimental research in psychology, psycholinguistics, and cognitive science has discovered and examined numerous psychological constraints on human information processing. The tendency to concentrate on (a) present, rather than missing cues, (b) the current focus of attention, and (c) recent information are three examples. Short term memory limitations provide a fourth constraint. This paper shows that psychological constraints such as these can be used effectively as domain-independent sources of bias to improve learning. We first show that cognitive biases can be automatically and explicitly encoded into a baseline training instance representation. We then investigate the related problems of cognitive bias interaction and cognitive bias selection, and compare two selection methods that make varying assumptions about the independence of individual component biases. Finally, the paper shows that performance of a nearest-neighbor case-based learning algorithm on a natural language task improves as more cognitive biases are explicitly encoded into the baseline instance representation.

1 Introduction

Inductive concept acquisition has always been a primary interest for researchers in the field of machine learning (see, e.g., Michalski et al. [1983], Michalski et al. [1986], and Michalski and Kodratoff [1990]). As a result, there are a large number of successful inductive learning systems (e.g., C4.5[Quinlan, 1992], COBWEB [Fisher, 1987], ID3 [Quinlan, 1986], UNIMEM [Lebowitz, 1987]). Independently, psychologists, psycholinguists, and cognitive scientists have examined the effects of numerous psychological limitations on human information processing. However, despite the fact that concept learning is a basic cognitive task, most machine learning systems for concept acquisition do not exploit cognitive processing limitations that constrain learning.

This paper shows that cognitive processing limitations can be used effectively as domain-independent sources of bias to constrain and improve learning. We first describe how cognitive biases can be automatically and explicitly encoded into a training instance representation. In particular, we use a simple k-nearest neighbor case-based learning algorithm and focus on a single learning task from the field of natural language processing (NLP). After presenting a baseline instance representation for the task, we modify the representation in response to four cognitive biases:

1. the tendency to concentrate on present, rather than missing cues,
2. a focus of attention bias,
3. the tendency to rely on the most recent information, and
4. short term memory limitations.

These modifications to the instance representation either *directly* or *indirectly* change the feature set used to describe all instances. Direct changes to the representation are made by adding or deleting features; indirect changes modify a weight associated with each feature. In 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. Unlike feature sets that have been reverse-engineered for a specific domain task, this approach to feature set specification relies on domain-independent psychological results.

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. Therefore, the paper next evaluates the effects of simultaneously applying multiple cognitive biases to a baseline instance representation. We find that a number of bias combinations outperform their individual component biases. In some cases, this occurs even when incorporation of the individual bias degrades system performance.

Finally, the paper investigates two methods for bias selection that make varying assumptions about the independence of individual processing limitations. The first approach to bias selection uses the training data to determine empirically which individual cognitive biases improve performance for a learning task and then specifies an instance representation that combines the best-performing variations of these biases. This method assumes that there will be no bias interaction. The second approach makes no assumptions about cognitive bias interactions and instead exhaustively evaluates all combinations of the available biases. Results of our experiments on cognitive bias selection show that the accuracy of the learning algorithm improves as the baseline representation incorporates increasingly more cognitive biases.

The next section introduces the natural language learning task used throughout the paper. It is followed by a description of the baseline instance representation (Section 3.1) and the method used to incorporate independently each of the four cognitive biases (Sections 3.2 - 3.5). Section 4 begins the evaluation of cognitive bias interaction, and Section 5 describes and evaluates the approaches to bias selection briefly outlined above.

2 A Natural Language Learning Task: Finding the Antecedents of Relative Pronouns

The task for our machine learning system is:

> Given a sentence with the relative pronoun "who," learn to recognize the phrase or phrases that represent the relative pronoun's antecedent.

Given the sentence in Figure 1, for example, the system should recognize that "the boy" is the antecedent of the relative pronoun because "who" refers to "the boy." Finding the antecedents of relative pronouns is a crucial task for natural language systems because the antecedent must be carried to the subsequent clause where it implicitly fills the actor role.[1] The semantic interpretation of the sentence in Figure 1, for example, should include the fact that "the boy" is the actor of "won" even though that phrase does not appear in the embedded clause "who on the award." To make this inference, a natural language system must associate "the boy" with "who" and then implicitly carry the constituent across the clause boundary to its correct position in the embedded clause.

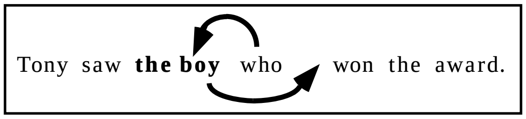


Figure 1: Understanding Relative Clauses

Although finding relative pronoun antecedents seems a simple enough task, there are many factors that make it difficult:[2]

- **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 Table 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 of the preceding clause. 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. The antecedent head may be very distant from the relative pronoun (e.g., S4).

[1] In practice, the antecedent of "who" sometimes fills semantic roles other than the actor, e.g., in passive verb constructions.

[2] Locating the implicit position that the antecedent fills in the embedded clause is a separate, but equally difficult problem, and will not be discussed here. See Cardie and Lehnert [1991] for a solution to this "gap-finding" problem that is consistent with the work presented here.

- **The antecedent may be a conjoined noun phrase.** In S5, for example, the antecedent of “who” is a conjunction of three phrases rather than the simple noun phrase “Shawn.”

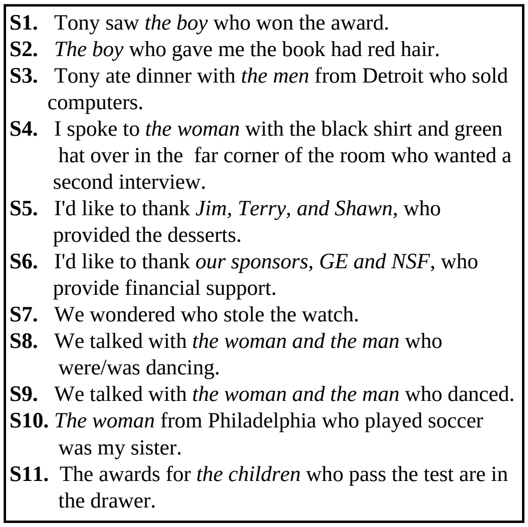
- **There may be more than one semantically valid antecedent.** In S6, the antecedent of “who” is either “our sponsors,” its appositive “GE and NSF,” or the entire phrase “our sponsors, GE and NSF.”

- **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.

- **Disambiguation of the relative pronoun may depend on information in the embedded clause.** In S8, for example, the antecedent of “who” is either “the man” or “the woman and the man,” depending on the number of the embedded clause verb.

- **Sometimes, the antecedent is truly ambiguous.** For sentences like S9, the real antecedent depends on the surrounding context.

- **Locating the antecedent requires the assimilation of both syntactic and semantic knowledge.** The syntactic structure of the clause preceding “who” in sentences S10 and S11, for example, is identical (a noun phrase followed by a prepositional phrase). The antecedent in each case is different, however. In S10, the antecedent is the subject, “the woman.” In S11, the antecedent is the prepositional phrase modifier, “the children.”



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 sponsors, GE and NSF*, who provide financial support.
S7. We wondered who stole the watch.
S8. We talked with *the woman and the man* who were/was dancing.
S9. We talked with *the woman and the man* who danced.
S10. *The woman* from Philadelphia who played soccer was my sister.
S11. The awards for *the children* who pass the test are in the drawer.

Table 1: Antecedents of “who”

Despite these ambiguities, we will describe how a machine learning system can learn to locate the antecedent of “who” given a description of the clause that precedes it. The system learns, in effect, to recognize the “relative pronoun antecedent” concept. More importantly, we will show that performance of the learning system improves as the instance description includes increasingly more cognitive limitations and cognitive biases.

The next section first describes the baseline instance representation developed for the relative pronoun antecedent problem. It then shows how each of four cognitive biases can be incorporated into the instance representation and measures the effects of each bias on relative pronoun antecedent prediction. Sections 4 and 5 address the problems of cognitive bias interaction and cognitive bias selection for the relative pronoun antecedent task.

3 Incorporating Cognitive Biases

Very generally, each training instance contains a list of attribute-value pairs that encode the context in which a relative pronoun (RP) is found. Although our approach to RP resolution is general enough to handle any relative pronoun, the work described here will concentrate solely on finding antecedents for "who." Each training instance is also annotated with a single class value that describes 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 baseline instance representation for the RP antecedent problem was designed with two constraints in mind. First, the instance representation should be one that is suitable for the sentence analyzer that creates the instances. In all our experiments, we use the CIRCUS sentence analyzer [Lehnert, 1990] to process training and test sentences and to create the associated instances. Although it is not necesssary to understand the details of the system, CIRCUS exhibits a number of processing characteristics that are relevant to the design of the instance representation for the RP task:

- CIRCUS recognizes phrases as it finds them in its left-to-right traversal of a sentence.
- CIRCUS recognizes major constituents like the subject, verb, and direct object.
- CIRCUS makes no immediate decisions on structural attachment. In particular, it does not handle conjunctions or appositives, or perform prepositional phrase attachment.
- CIRCUS uses one or more semantic features to describe every noun and adjective in its lexicon. E.g., "mayor" is tagged as human, "ELN" is tagged as an organization, and the noun "murder" is tagged as an attack.
- CIRCUS gives special attention to the phrase most recently recognized.
- CIRCUS treats punctuation marks in much the same way as it treats words.
- CIRCUS has the ability to examine and report its state at any point during the parse.

There is a second constraint on the baseline representation for the RP antecedent problem. As seen in the examples of Table 1, the antecedent of "who" usually appears as one or more phrases in the clause preceding the relative pronoun. For this reason, the instance representation should include a feature for every constituent from the clause that precedes "who." The baseline instance representation described next incorporates this problem constraint and makes use of the prominent parser characteristics listed above.

3.1 The Baseline Representation

Throughout the paper, we employ a simple case-based, or "instance-based" [Aha *et al.*, 1991], learning algorithm for the RP antecedent task. During the training phase, the

CIRCUS sentence analyzer generates one case for each occurrence of "who" that appears in a set of training sentences and stores them in a case base. After training, the system uses the case base to predict the antecedent of "who" in new contexts. Given a new RP instance to classify, the case retrieval algorithm compares the test case to those stored in the case base, finds the most similar training case, and then uses it to predict the antecedent for "who" in the novel example. In the baseline instance representation, each training and test case contains three sets of attribute-value pairs that describe the context in which the relative pronoun was found:

1. A set of **constituent** attribute-value pairs, one for every phrase in the clause preceding the relative pronoun, where the

   attribute describes the the *syntactic class and position* of the phrase as it was encountered by the parser, and the

   value provides the phrase's semantic classification.

2. An attribute-value pair that represents *semantic* information associated with the most recent phrase, where the

   attribute is **most-recent**, and the

   value provides the semantic classification of the phrase's head noun.

3. An attribute-value pair that denotes *syntactic* information for the phrase or punctuation mark immediately preceding "who," where the

   attribute is **syn-type** (for "syntactic type"), and the

   value provides either the phrase's syntactic class (e.g., noun phrase, verb, prepositional phrase) or the type of the most recent punctuation mark (e.g., comma, semicolon), whichever is closest to the relative pronoun.

Consider, for example, the sentences in Figure 2. In S1, there are two phrases in the clause preceding "who." In the training instance for S1, therefore, there is a feature for each phrase. We represent "the man" with the attribute-value pair *(s human)* because it is the subject of the sentence and the noun "man" is human. We represent "from Oklahoma" with the pair *(s-pp1 location)* because it is the first prepositional phrase that follows the subject and "Oklahoma" is a location.[3] Because "from Oklahoma" is also the most recently recognized phrase with respect to the onset of "who," the value of the **most-recent** feature is also *location*. Finally, because a punctuation mark immediately precedes "who," *comma* is the value associated with the **syn-type** feature. (If S1 had contained no comma, then the value of **syn-type** would have been *prep-phrase*.)

When clauses contain conjunctions and appositives, each phrase in the construct is labeled separately. In S2, for example, the real direct object of "thank" is the conjunction "Nike and Reebok." In CIRCUS, and therefore in our instance representation, however, "Nike" is tagged as the direct object *(do)* and "Reebok" as the first noun phrase that follows the direct object *(do-np1)*. Because verb phrases have no semantic features in CIRCUS, *t* is used as the value for the verb (**v**) attribute. For S2, the most recent phrase is the *proper-name* "Reebok," and a *comma* precedes the relative pronoun.

[3] All noun phrases are described by one of seven general semantic features: human, proper-name, location, entity, physical-target, organization, and weapon. 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.

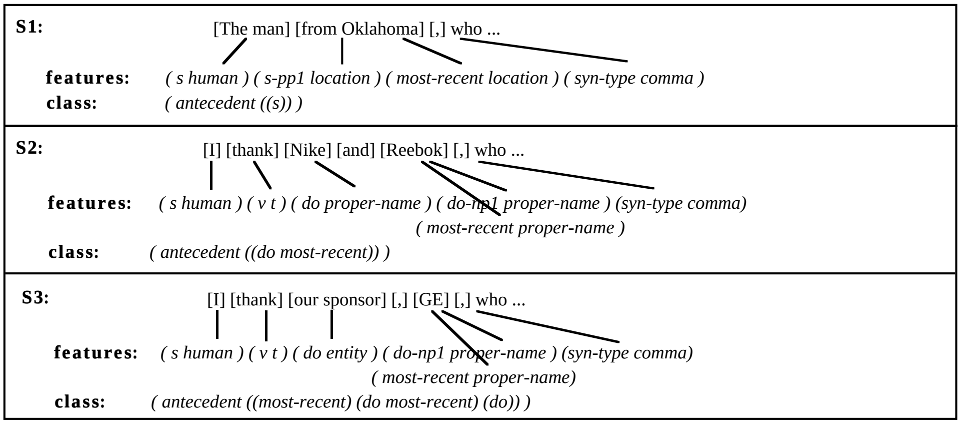

S1:
[The man] [from Oklahoma] [,] who ...
features: (s human) (s-pp1 location) (most-recent location) (syn-type comma)
class: (antecedent ((s)))
S2:
[I] [thank] [Nike] [and] [Reebok] [,] who ...
features: (s human) (v t) (do proper-name) (do-np1 proper-name) (syn-type comma)
(most-recent proper-name)
class: (antecedent ((do most-recent)))
S3:
[I] [thank] [our sponsor] [,] [GE] [,] who ...
features: (s human) (v t) (do entity) (do-np1 proper-name) (syn-type comma)
(most-recent proper-name)
class: (antecedent ((most-recent) (do most-recent) (do)))

Figure 2: Baseline Instance Representation

Every training instance is also annotated with class information — information about where to find the correct antecedent of "who" for the current example. We provide class information as a list of the constituent attributes that represent the location of the antecedent or *(none)* if no antecedent is required. In S1, for example, the antecedent of "who" is "the man." Because this phrase is represented as the constituent pair *(s human)*, the value of the antecedent class is *(s)*. If the antecedent were actually "from Oklahoma," however, then the class information would refer to the more general *most-recent* constituent rather than the semantically equivalent, but more specific, *s-pp1* constituent attribute. 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. Again, the more general *most-recent* label is used to denote antecedent phrases that immediately precede the relative pronoun. In S2, for example, because "who" refers to the conjunction "Nike and Reebok," the antecedent is described as *(do most-recent)*. S3 shows yet another variation of the antecedent. In this example, an appositive creates three semantically equivalent antecedents, all of which become part of the antecedent class information:

"GE"	*(most-recent)*
"our sponsor"	*(do)*
"our sponsor, GE"	*(do most-recent)*

As described earlier, the CIRCUS sentence analyzer generates one case in the baseline representation for each occurrence of "who" that appears in a set of training sentences. It should be noted that training instances are generated automatically from unrestricted texts as a side effect of parsing. Only the correct antecedent must be specified by a human supervisor via a menu-driven interface that displays the antecedent options. All training instances are stored in a case base. After training, given a new instance to classify described in terms of a set of **constituent** attribute-value pairs, a **most-recent** feature, and a **syn-type** feature, the case retrieval algorithm compares the test case to those stored in the case base, finds the most similar training case, and then uses it to predict the antecedent for "who" in the novel example.

Given the test case in Figure 3, for example, assume that the case retrieval algorithm retrieves an instance that specifies the direct object (*do*) as the location of the antecedent. Then the current *do* constituent — "the hardliners" — would be chosen as the antecedent in the novel case. Sometimes, however, the retrieved case may list more than one option as the antecedent. In these cases, we choose the first option whose constituents overlap with those in the current example.[4] The next section describes the case retrieval algorithm in more detail and presents the results of experiments using the baseline instance representation.

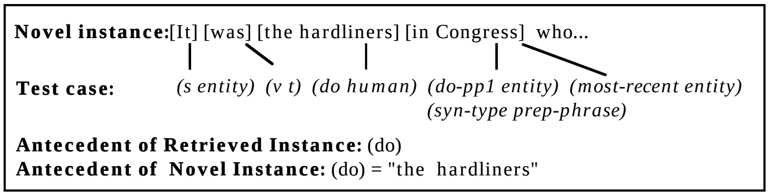

Novel instance:[It] [was] [the hardliners] [in Congress] who...
Test case:
(s entity) (v t) (do human) (do-pp1 entity) (most-recent entity)
(syn-type prep-phrase)
Antecedent of Retrieved Instance: (do)
Antecedent of Novel Instance: (do) = "the hardliners"

Figure 3: Instance Retrieval

3.2 Case Retrieval and the Baseline Results

In this section, we describe the case retrieval algorithm and the experiments that evaluate the baseline instance representation for the RP antecedent task. Because of its relevance to both case retrieval and the baseline instance representation, however, we first discuss the absence of a uniform set features for cases. Although all instances created using the baseline representation described above have **most-recent** and **syn-type** attributes, there is no guarantee that two instances will have additional attributes in common. S1 and S2 in Figure 2, for example, only have one additional attribute in common (**s**). The same is true for S1 and S3. S2 and S3, on the other hand, have exactly the same set of attributes (although the values of those attributes do not all agree). If the case retrieval algorithm determines case similarity by simply counting the number of features in a training case that match those in the test case, then two training cases that match the same number of test case features will achieve the same similarity score regardless of the percentage of training case features correctly matched. Consider the following scenario:

Training case 1:	((s attack) (s-pp1 human) (v t) (do organization) (most-recent human) (syn-type comma))
Training case 2:	((s attack) (s-pp1 human) (most-recent human) (syn-type comma))
Test case:	((s attack) (s-pp1 human) (most-recent human) (syn-type comma))

The case retrieval algorithm described above assigns the same similarity score to training case 1 and training case 2. This occurs even though training case 2 is an exact match for the test case.

The case retrieval algorithm used in all of our experiments will require a systematic mechanism for handling this problem. For a solution, we can look to a first cognitive processing limitation — the *present vs. missing cues bias*. Psychological models of concept acquisition actually support two views of case similarity. A number of incremental models of concept acquisition and category construction update the predictive value of individual features after the arrival of each new instance, but do so *only for features appearing in the incoming instance* (e.g., Bourne et al. [1976], McDonald and MacWhinney [1991]). These models implement the present cues bias because they focus on features that appear in the current instance. Other models, however, implicitly assume that all instances will be described in terms of the same set of attributes (e.g., Ahn and Medin [1992], Medin et

[4] For a description of better, but more complicated heuristics, see Cardie [1992a].

al. [1987]). These models implement the missing cues bias because they use a normalized set of features to describe each instance.

The system incorporates the present vs. missing cues bias as part of the baseline representation two steps. First, it directly modifies the current instance representation by describing all instances in terms of a normalized set of features:

1. Derive a *normalized feature set* by keeping track of every attribute that occurs in the training instances.
2. Augment the training and test instances to include every feature of the normalized feature set, filling in a *nil* value if the feature is irrelevant for the particular instance.

Second, the system indirectly modifies the instance representation by associating with each feature a weight that indicates the importance of that feature in determining case similarity.

Using this expanded baseline representation, we can employ the following weighted 1-nearest neighbor (1-nn) case retrieval algorithm to find the training case most similar to a test case:

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

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

   where N is the normalized feature set, 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 the training case, and $match(a, b)$ is a function that returns 1 if a and b are equal and 0 otherwise. (The procedure for initializing the weight vector will be discussed below.)
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.

To implement the present cues bias, we set w_{N_i} to 0 for all features from the normalized feature set that have *nil* values in the test case, and to 1 for the remaining features, allowing matches only on features in the original test case to have an effect on the total returned by the similiarity metric. To implement the missing cues bias, we set w_{N_i} to 1 for all attributes in the normalized feature set, allowing matches across all features to have an effect on the total returned by the similiarity metric and treating present and missing cues as equally important in determining case simliarity. We also experiment with a similarity weighting procedure that allows partial matches on missing features by setting w_{N_i} to a value greater than 0 and less than 1 for all features missing from the original test case, and to 1 for the remaining features.

We tested four variations of the expanded baseline representation. In each variation, we set (1) w_{N_i} to one of 0, 0.2, 0.5, or 1 for all features in the normalized feature set that were missing from the original test case, and (2) w_{N_i} to 1 for all remaining features. The variations were tested using the weighted 1-nn case retrieval algorithm and a 10-fold cross validation testing scheme. Specifically, CIRCUS first generates cases for all 241 examples of "who" from three sets of 50 texts from the MUC-3 corpus.[5] Next, the 241 examples are

[5] The MUC-3 corpus consists of 1500 texts (e.g., newspaper articles, TV news reports, radio broadcasts) containing information about Latin American terrorism and was developed for use in the Third Message Understanding System Evaluation and Message Understanding Conference [Sundheim, 1991]. Texts vary

randomly partitioned into ten non-overlapping segments of 24 instances. (One instance appears in none of the ten segments.) In each of ten runs, we reserve the instances in one segment for testing and use the remaining 217 instances for training. For each test case, we invoke the 1-nn case retrieval algorithm to predict the antecedent. Results are averaged across the ten runs.

The results of the baseline experiments are shown in Table 2 with the accuracies of (1) a system that always chooses the most recent constituent as the antecedent, and (2) a set of hand-crafted heuristics for RP resolution that were developed for use in the MUC-3 performance evaluation.[6]

In summary, the representation that treats present cues and missing cues as equally important (i.e., missing cues weight = 1) performs best among the baseline representations, but its increases over the other baselines are not statistically significant. Chi-square significance tests indicate that none of the baseline representations performs significantly better or worse than the default rule. (All statements of statistical significance are at the 95% confidence level unless otherwise noted.) Two versions of the baseline representation (missing cues wt = 0, 0.5), however, perform significantly worse than the hand-coded heuristics.

Missing Cues Wt = 0	**Missing Cues Wt = 0.2**	**Missing Cues Wt = 0.5**	**Missing Cues Wt = 1**	**Default Rule**	**Hand-Coded Heuristics**
74.58	76.25	75.42	78.75	75.0	80.7

Table 2: Results for the Baseline Representation (% correct)

In the following sections, 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 RP antecedents. Unless otherwise mentioned, the 1-nn case-based algorithm will be used for all experiments, and the results shown will be 10-fold cross validation averages. In particular, we emphasize that the same ten training and test set combinations as the baseline experiments will be used in all subsequent experiments. This procedure assures us that differences in performance are not attributable to the random partitions chosen for the test set.

3.3 Incorporating the Subject Accessibility Bias

A number of studies in psycholinguistics have noted the special importance of the first item mentioned in a sentence. In particular, it has been shown that the accessibility of the subject of a sentence remains high even at the end of a sentence [Gernsbacher *et al.*, 1989]. This *subject accessibility bias* is an example of a more general *focus of attention bias*. In vision learning problems, for example, the brightest object in view may be a highly accessible object for the learning agent; in aural tasks, very loud or high-pitched sounds may be highly accessible. We incorporate the subject accessibility bias into the baseline

from a paragraph or two to over a page in length. Approximately 25% of the sentences in the corpus contain one or more wh-words (i.e., who, whom, which, whose, where, when, why). The relative pronoun "who" occurs in approximately 1 out of every 10 sentences.

[6] The UMass / MUC-3 system that used these heuristics for RP resolution posted the highest score of 15 sites participating in the MUC-3 performance evaluation.

representation by increasing the weight associated with the constituent attribute (**s**) that represents the subject of the clause preceding the relative pronoun whenever that feature is part of the normalized feature set. Table 3 shows the effects of allowing matches on the **s** attribute to contribute 2, 5, 7, and 10 times as much as they did in the corresponding run of the baseline representation. Weights on the **s** attribute were chosen arbitrarily.

Missing Cues Wt	**Baseline**	**Subject Weight = 2**	**Subject Weight = 5**	**Subject Weight = 7**	**Subject Weight = 10**
0	74.58	73.75	72.92	73.75	72.50
0.2	76.25	75.00	74.17	73.75	73.33
0.5	75.42	74.58	75.00	73.33	71.67
1	78.75	78.33	77.08	75.42	76.25

Table 3: Results for the Subject Accessibility Bias Representation (% correct)

Results indicate that incorporation of the subject accessibility bias never improves performance of the learning algorithm, although dips in performance are never statistically significant. At first these results may seem surprising; however, the baseline representation produced by the sentence analyzer 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**). It may be that the original encoding of the bias is adequate or that more modifications to the baseline representation are required before the subject accessibility bias can have an additional positive effect on the learning algorithm's ability to find RP antecedents.

3.4 Incorporating the Recency Bias

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

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

We translate this recency bias into representational changes for the training and test instances in two ways. The first is a direct modification to the instance representation, and the second modifies the weights to indicate a constituent's distance from the relative pronoun. In the first approach, we label the **constituent** attribute-value pairs by their position relative to relative pronoun. This establishes a right-to-left labeling rather than the left-to-right labeling that the baseline representation incorporates. In Figure 4, for example, "in Congress" receives the attribute *pp1* because it is a prepositional phrase one position to the left of "who." Similarly, "the hardliners" receives the attribute *np2* because it is a noun phrase two positions to the left of "who." The right-to-left ordering yields a different feature set and, hence, a different instance representation. For example, the right-to-left labeling tags the antecedents in both of the following sentences with the same attribute (i.e., *pp2*):

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

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

Sentence:[It] [was] [the hardliners] [in Congress] who...
Baseline Representation: *(s entity) (v t) (do human) (do-pp1 entity) (most-recent entity)*
(syn-type prep-phrase) (antecedent ((do)))
Right-to-Left Labeling: *(s entity) (v t) (np2 human) (pp1 entity) (most-recent entity)*
(syn-type prep-phrase) (antecedent ((np2)))

Figure 4: Incorporating the Recency Bias Using a Right-to-Left Labeling

In the second approach to incorporating the recency bias, we increment the weight associated with a constituent attribute as a function of its proximity to the relative pronoun (see Table 4). The feature associated with the constituent farthest from the relative pronoun receives a weight of one, and the weights are increased by one for each subsequent constituent. All features added to the instance as a result of feature normalization (not shown in Table 4) receive a weight of one. In addition, we assign the maximum weight to both the **most-recent** and **syn-type** features.

Phrase	**Feature**	**Baseline Weight**	**Recency Weight**
It	s	1	1
was	v	1	2
the hardliners	do	1	3
in Congress	do-pp1	1	4
	most-recent	1	5
	syn-type	1	5
who...			

Table 4: Incorporating the Recency Bias by Modifying the Weight Vector (for the sentence “It was the hardliners in Congress, who...”)

The results of experiments that use each of the recency representations separately and in a combined form are shown in Table 5. To combine the two implementations of the recency bias, we first relabel the attributes of an instance using the right-to-left labeling and then initialize the weight vector using recency weighting procedure described above. The table shows that the recency weighting implementation tends to degrade prediction of RP antecedents; the right-to-left labeling and combined representations clearly improve performance. Significant differences in performance with respect to the corresponding baseline representation are indicated in the table by $*$'s. The results of the recency bias are outstanding for an additional reason: like the subject accessibility bias, the parser provides a built-in recency bias because it represents the constituent that precedes the relative pronoun up to three times in the baseline representation — as a **constituent** feature, the **most-recent** feature, and the **syn-type** feature. The last column in Table 5 shows the performance of the baseline representation when this built-in bias is removed by discarding the **most-recent** and **syn-type** features and disallowing references to them in the **antecedent** class value.

Missing Cues Wt	Baseline	R-to-L Labeling	Recency Weighting	R-to-L + RecWt	Baseline w/o Recency
0	74.58	78.75	73.75	80.83*	63.3
0.2	76.25	79.17	75.83	80.00	69.2
0.5	75.42	81.25*	75.42	79.58	69.6
1	78.75	80.00	76.25	80.83	67.5

Table 5: Results for the Recency Bias Representations (% correct) (* indicates significance with respect to the baseline results at the 95% level, $p = 0.05$.)

It is interesting that the combined recency bias outperforms the right-to-left labeling for three out of four baseline variations despite the fact that the recency weighting tends to lower the accuracy of RP antecedent prediction when used alone. These results imply that the representation of the local context of the relative pronoun provided by the right-to-left labeling is critical for finding antecedents. The disappointing performance of the recency weighting representation, on the other hand, may be caused by (1) its lack of such a representation of local context, and (2) its bias against antecedents that are distant from the relative pronoun (e.g., "...to help especially *those people* living in the Patagonia region of Argentina, who are being treated inhumanely..."). Nineteen of the 241 instances have antecedents that include the subject of the preceding clause.

3.5 Incorporating the Short Term Memory Bias

Psychological studies have determined that people can remember at most seven plus or minus two items at any one time [Miller, 1956]. More recently, Daneman and Carpenter [1983,1980] show that working memory capacity affects a subject's ability to find the referents of pronouns over varying distances. 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.

The baseline representation does not necessarily make use of short term memory (STM) limitations, however. All of the baseline variations describe instances using the normalized feature set, which contains an average of 38.8 features taken over the all partitions of the 10-fold cross validation. The present cues bias variation, however, which ignores features missing from the original test case, actually employs the STM bias whenever the number of features in the original test case is small enough.[7] Even in these cases, previous STM studies do not explicitly state what the short term memory limit should be — it varies from five to nine depending on the cognitive task and depending on the size and type of

[7] The average number of features in an unnormalized test case is 5.4.

the “chunks” that have to be remembered. In addition, the STM bias alone does not state which features to keep and which to discard.

To more rigorously apply the STM bias to the baseline representations, we let n represent the STM limit and, in each of five runs, set n to one of 5, 6, 7, 8, or 9. Then, for each test case, the system randomly chooses n features from the normalized feature set, sets the weights associated with those features to one, and sets the remaining weights to zero. This effectively discards all but the n selected features from the instance representation. For the present cues bias variation (i.e., missing cues weight = 0), however, the n features are chosen from among the original test case features.

Missing Cues Wt	**Baseline**	**STM Limit = 5**	**STM Limit = 6**	**STM Limit = 7**	**STM Limit = 8**	**STM Limit = 9**
0	74.58	76.67	75.83	74.58	77.50	75.00
0.2	76.25	78.33	74.17	76.25	75.83	75.00
0.5	75.42	73.33	74.58	75.00	77.50	73.33
1	78.75	76.25	74.58	72.50*	75.00	76.25

Table 6: Results for the Short Term Memory Bias Representation (% correct) (* indicates significance with respect to the baseline results at the 95% level, $p = 0.05$.)

Results for the STM bias representations are shown in Table 6. Although none of the increases is statistically significant, the STM bias improves performance for four out of five values of n when the present cues bias is in effect (i.e., when missing cues wt = 0). Because the average number of features in an unnormalized RP case is 5.4, the STM bias actually affects a relatively small number of cases. In general, however, the STM bias degrades the ability of the system to predict RP antecedents. This is not surprising given that the current implementation of the STM 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 when it is combined with cognitive biases that provide more feature relevancy information.

3.6 Discussion

It should be emphasized that modifications to the baseline instance representation in response to each of the individual cognitive biases are performed automatically, subject to the constraints provided in Table 7. The user need only specify the bias name and any associated parameters upon invocation of the nearest-neighbor learning algorithm. The ease of incorporating these biases makes it possible to apply them to learning tasks in other domains. The right-to-left labeling implementation of the recency bias, for example, can be applied to any learning task for which (1) there is a temporal ordering among the features, and (2) this ordering is denoted in the attribute names. To invoke this bias, the user must supply a function that maps the original attribute names to new attribute names.

Based on the results in this section, we can conclude that, compared to the corresponding baseline results (1) the subject accessibility bias does not improve the accuracy of RP antecedent prediction, (2) the recency weighting bias degrades performance unless used in combination with the right-to-left labeling bias, (3) the right-to-left labeling bias can significantly improve antecedent prediction, (4) combining both recency bias implementations can significantly improve performance, and (5) the short term memory bias improves

Bias	**Assumptions**	**Parameters**
Present vs. Missing Cues	Instances have different sets of known features	Weight associated with missing features
Focus of Attention (subject accessibility)	None	Weight factor, attribute associated with object of focus, e.g., the subject
Recency (r-to-l labeling)	Attribute names indicate recency	Function mapping original attribute names to new attribute names
Recency (recency weighting)	Attributes in original instance provided in inverse recency order	None
Short Term Memory Limitations	None	STM threshold

Table 7: Cognitive Bias Modifications

performance when used with the present cues bias variation of the baseline. Table 8 provides a summary of the best-performing variation of each instance representation tested in this section. While two of the recency bias representations outperform the best baseline variation (78.75% correct) and the hand-coded rules (80.7% corrrect), neither increase is statistically significant.

The next section examines the interactions that exist between pairs of cognitive biases.

Cognitive Bias	**Parameters**	**% Correct**
Present vs. Missing Cues (baseline)	missing cues wt=1	78.75
Subject Accessibility	subject wt=2 missing cues wt=1	78.33
Recency (R-to-L Labeling)	missing cues wt=0.5	81.25
Recency (Recency weighting)	missing cues wt=1	76.25
Recency (R-to-L, RecWt Combination)	missing cues wt=0	80.83
Short Term Memory	STM limit=5 missing cues wt=0.2	78.33

Table 8: Individual Cognitive Bias Summary

4 Cognitive Bias Interactions

Two experiments from the previous section showed that cognitive bias interactions may play a large role in selecting which biases to apply to a particular learning task. First, the merged recency bias representation performed very well despite the rather dismal performance of the recency weighting representation (Section 3.4). Second, the short term memory bias worked very differently when used in combination with the present cues bias and the missing cues bias (Section 3.5).

In each of the following sections, we investigate additional cognitive bias interactions by modifying the baseline representation in response to all pairs of cognitive biases. While the exact method used to merge component biases will be specified for each pair below, the following steps generally describe the merging procedure, given a set of cognitive biases to apply to a baseline instance representation:

1. First, incorporate any bias that relabels attributes.
2. Then, incorporate biases that modify feature weights by adding the weight vectors proposed by each bias.
3. Finally, incorporate biases that discard features, but give preference to those features assigned the highest weights in step 2.

In the sections below, the performance of each combined representation will be compared to the performance of the standalone representations and the baseline.

4.1 Combining the Subject Accessibility and Recency Biases

Recall that there are two ways to implement the recency bias — right-to-left labeling and recency weighting. In the following experiments, we combine the subject accessibility bias with each implementation of the recency bias.

4.1.1 Subject Accessibility and Right-to-Left Labeling

The first experiments combine the subject accessiblity and right-to-left labeling recency biases. Before performing these experiments, we predicted that combining these biases would not improve performance of the learning algorithm on the RP antecedent task for the following reasons: (1) the the subject bias performed poorly on its own, (2) the right-to-left bias performed very well on its own, and (3) a subject bias is already somewhat built into the right-to-left labeling representation, as in the original instance representation.

Merging the subject accessibility and right-to-left recency instance representations is straightforward as a matter of implementation: first incorporate the right-to-left labeling bias by relabeling the constituents with respect to the relative pronoun; then incorporate the subject accessibility bias by modifying the weight associated with the subject feature.

Results for the combined bias representation are shown in Table 9. Although in many cases the new representation performs significantly better than the corresponding subject accessibility representation (Table 3), it never outperforms (and occasionally significantly underperforms) the corresponding right-to-left labeling representation. These results confirm our initial predictions.

4.1.2 Subject Accessibility and Recency Weighting

The second experiments combine the subject accessibility and recency weighting biases. Because both biases modify the attribute weight vector, we can implement the combined bias by adding the weight vectors proposed by the individual biases. Although neither bias performed particularly well when used alone, we had noted two difficulties with the recency weighting representation — it lacked a good representation of the context local

Missing Cues Wt	Baseline	R-to-L Labeling	R-to-L, SubjWt = 2	R-to-L, SubjWt = 5	R-to-L, SubjWt = 7	R-to-L, SubjWt = 10
0	74.58	78.75	78.33	78.33	78.33	77.50
0.2	76.25	79.17	77.50	78.33	78.33	77.50
0.5	75.42	81.25	78.75	75.83*	76.25*	77.50
1	78.75	80.00	78.33	78.75	77.92	78.75

Table 9: Combining the Subject Accessibility and R-to-L Labeling Biases (% correct) (* indicates significance with respect to the right-to-left labeling results at the 95% level, $p = 0.05$.)

to the relative pronoun, and was biased against distant antecedents. Because the subject accessibility bias may help to alleviate the second problem, we expected the merged representation to perform slightly better than the standalone recency weighting representation.

This prediction was not supported by the results (Table 10). The combined bias representation consistently underperforms the corresponding recency weighting representation, subject accessibility representation (Table 3), and baseline representation. None of the differences in performance is significant. It appears that the subject accessibility bias alone does not compensate for the recency weighting bias deficiencies.

Missing Cues Wt	Baseline	Recency Weighting	RecWt, SubjWt = 2	RecWt, SubjWt = 5	RecWt, SubjWt = 7	RecWt, SubjWt = 10
0	74.58	73.75	74.17	72.92	72.08	72.50
0.2	76.25	75.83	74.58	72.92	72.50	73.33
0.5	75.42	75.42	75.42	73.75	72.08	73.75
1	78.75	76.25	76.25	73.33	73.75	73.33

Table 10: Combining the Subject Accessibility and Recency Weighting Biases (% correct)

4.1.3 Subject Accessibility and Combined Recency

Finally, we combine the subject accessibility and both recency biases by first establishing the right-to-left labeling of features and then adding together the weight vectors recommended by the recency weighting and subject accessibility biases. Here, we expected increases in performance using the merged representation because (1) the combined recency bias representation worked well in isolation and (2) the subject accessibility bias may increase the system's ability to find RP antecedents in the subject position. We predicted that these increases would be small, however, since some bias toward the subject is already encoded in the combined recency bias representation.

The results are shown in Table 11 and, for the most part, confirm these predictions. In general, the new instance representation underperforms the combined recency bias representation. However, it performs better than the recency bias representation for three out of four variations when the weight associated with the subject was set to two. Although none of the increases in performance with respect to the recency combination representation is significant, two of these variations perform significantly better than the associated baseline representation. The new representation also generally outperforms the subject accessibility representation (see Table 3) and, as indicated in Table 11, many of these increases are significant.

Missing Cues Wt	Baseline	Recency Combinbation	RecComb, SubjWt = 2	RecComb, SubjWt = 5	RecComb, SubjWt = 7	RecComb, SubjWt = 10
0	74.58	80.83	82.50**⋄⋄	80.00⋄⋄	78.33⋄	77.92⋄
0.2	76.25	80.00	79.58⋄	78.33	77.50	76.67
0.5	75.42	79.58	80.83*⋄⋄	77.92	77.50	77.08⋄
1	78.75	80.83	81.67	76.67	76.67	78.33

Table 11: Combining the Subject Accessibility, Right-to-Left Recency Labeling, and Recency Weighting Biases (% correct) (* and $\diamond$ indicate significance with respect to the corresponding baseline and subject accessibility representations (Table 3), respectively: $*, \diamond \rightarrow p = 0.05$ and $**, \diamond\diamond \rightarrow p = 0.01$.)

4.2 Combining the Subject Accessibility and STM Biases

This section describes the effects of combining the subject accessibility and STM biases. As described in Sections 3.3 and 3.5, both biases indirectly modify the instance representation by changing the weight vector used in the similarity metric. The subject accessibility bias changes the weight associated with the subject, and the STM bias randomly discards features by setting their weights to zero. To combine the biases, we first make the subject accessibility modifications and then apply the STM bias, but never allow the STM bias to discard the subject feature. Because the resulting instance representation varies very little from the original STM representation, we expected performance using the new representation to vary little from the STM bias results.

After running the 80 variations of the combined bias representation[8], we found that the new representation generally performed worse than the corresponding STM representation and achieved mixed results with respect to the corresponding subject accessibility representation. Only two variations of the new representation outperform the corresponding subject accessibility, STM bias, and baseline representations. These are graphed in Figure 5 with the results of the corresponding STM bias, baseline, and subject accessibility representations. None of the increases is statistically significant. Although results for combining the STM and subject accessibility biases are inconclusive, it appears that, with the correct parameter settings, the merged representation can improve performance of the learning algorithm.

4.3 Combining the STM and Recency Biases

Finally, we examine the interactions between the short term memory and recency biases.

4.3.1 STM and Right-to-Left Labeling

Merging the right-to-left labeling and STM representations is straightforward. We first incorporate the right-to-left recency bias by relabeling the attributes with respect to the relative pronoun and then incorporate the STM bias by randomly choosing the features to discard. Because the right-to-left labeling proved a good representation, but the STM bias

[8] For each of four baseline representations, we set the subject weight to one of 2, 5, 7, or 10, and the STM limit to one of 5, 6, 7, 8 or 9.

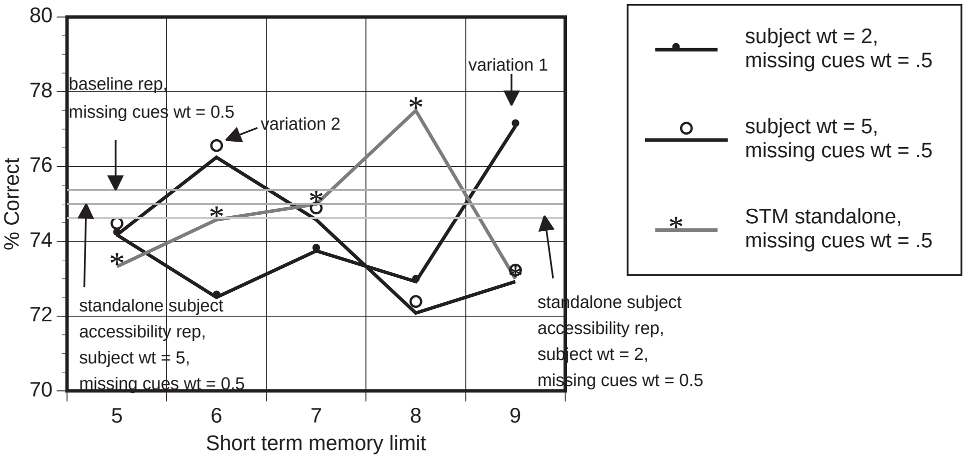


Figure 5: Combining the Subject Accessibility and STM Biases

still randomly discards features, we predicted that the new representation would perform better than the STM bias representation, but would not outperform the right-to-left labeling representation. The only exception might occur in variations in which the present cues bias was in effect because the original STM experiments (Section 3.5) determined that the STM bias had the greatest impact on performance in these cases.

Our experiments confirmed these predictions. For all trials in which the missing features bias was in effect (i.e., when missing cues wt = 0.2, 0.5, or 1), the combined representation underperformed the corresponding right-to-left bias (sometimes significantly). Results for the present cues bias variations are shown in Figure 6 and show that three out of five variations of the new representation (STM limit = 6, 7, and 8) perform better than the corresponding baseline, STM, and right-to-left labeling representations. These results indicate that the present cues, STM, and right-to-left labeling biases work well together.

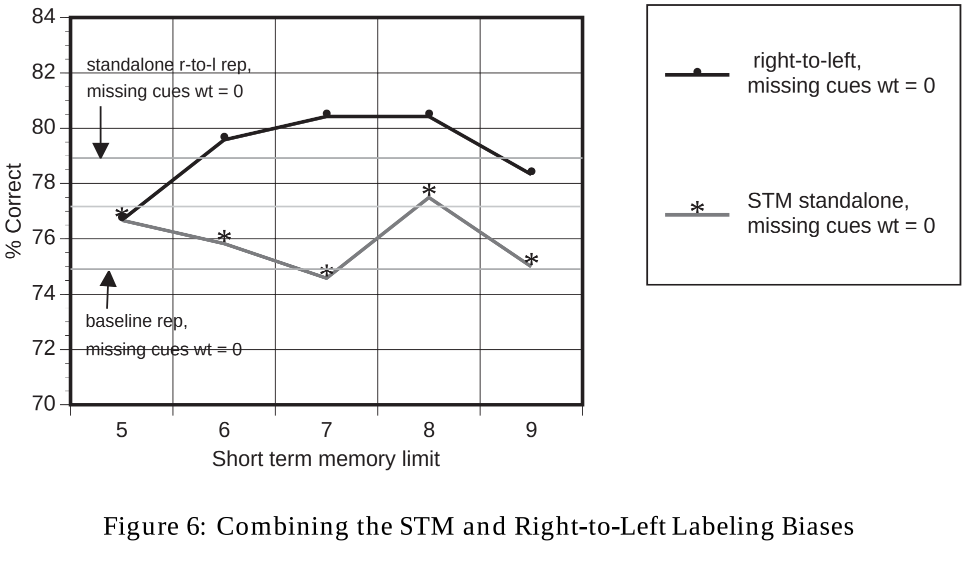


Figure 6: Combining the STM and Right-to-Left Labeling Biases

4.3.2 STM and Recency Weighting

Next, we combine the STM bias and recency weighting representations. This combination requires two modifications to the weights associated with the normalized feature vector. We first rank the features according to the recency weighting bias and then keep the n best features (where n is the STM limit), choosing randomly in case of ties. When the present cues bias is active, the new representation approximates the recency weighting representation as the STM limits decreases. In addition, the STM bias solves neither of the aforementioned recency weighting bias deficiencies — lack of a good representation of local context and a bias against distant antecedents. As a result, we predicted that the new representation would not outperform the standalone representations.

For the most part, our experiments confirmed these predictions. In general, the new representation performed slightly worse than the corresponding recency representation and achieved mixed results when compared to the corresponding STM represenation. With one exception (STM limit = 5, missing cues weight = 0), none of the new variations outperformed both corresponding standalone representations as well as the corresponding baseline.

4.3.3 STM and Recency Combination

Finally, we combine the STM representation with both implementations of the recency bias by applying the right-to-left labeling, ranking the features according to the recency weighting, and then keeping the n best features (where n is the STM limit). Ties are broken randomly. Because the combined recency bias representation worked well on its own and because the STM bias discards features that are distant from the relative pronoun and rarely included in the antecedent, we predicted that the new representation would perform rather well.

Experiments clearly verified these predictions. Each of the 20 variations of the combined representation outperforms its STM counterpart as well as the corresponding baseline representation. More importantly, 16 of the 20 variations also outperform the corresponding recency combination. In particular, increases in performance are most dramatic when the STM limit is five or six. Results for these variations are shown in Figure 7 with results from two of the best-performing STM runs.

4.4 Discussion

In this section, we have examined the interactions that exist between pairs of cognitive biases in order to identify those biases that work well togther. Experiments showed that a number of representations that merged two cognitive biases outperformed the corresponding standalone representations as well as the associated baseline representation. These variations are summarized in Table 12. By modifying the baseline representation in response to two, rather one, cognitive bias, the best performance of the case-based learning algorithm increases from 81.25% correct (right-to-left labeling bias) to 83.33% correct (STM and combined recency biases) for the RP antecedent task. While this increase is not statistically significant, the improvement with respect to the best baseline (78.75% correct) is statistically significant at the 95% confidence level.

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 other required paramters.

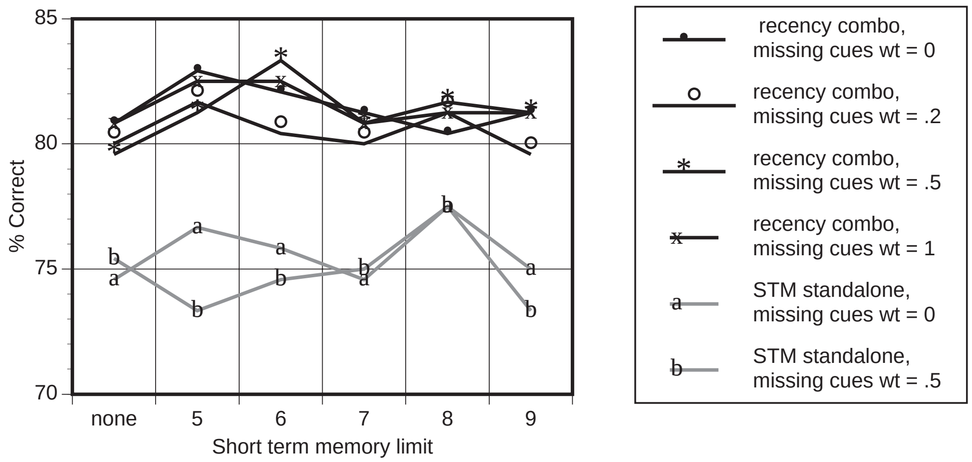


Figure 7: Combining the STM and Recency Combination Biases

Bias I	Bias II	Missing Cues Wt	Bias I (% correct)	Bias II (% correct)	Baseline (% corrrect)	Bias I + Bias II (% correct)
subj access	recency					
subj wt=2	combination	0	73.75	80.83	74.58	82.50
		0.5	74.58	79.58	75.42	80.83
		1	78.33	80.83	78.75	81.67
STM	r-to-l					
limit=6	recency	0	75.83	78.75	74.58	79.58
limit=7		0	74.58	78.75	74.58	80.42
limit=8		0	77.50	78.75	74.58	80.42
STM	recency					
limit = 5	weighting	0	76.67	73.75	74.58	77.50
[best 4]						
STM	recency					
limit=6	combination	0.5	74.58	79.58	75.42	83.33
limit=5		0	76.67	80.83	74.58	82.92
limit=6		1	74.58	80.83	78.75	82.50
limit=5		1	76.25	80.83	78.75	82.50

Table 12: Summary of Cognitive Bias Combinations That Performed Well

5 Cognitive Bias Selection

One problem that remains is to select the combination of cognitive biases that will achieve the best performance for a particular learning task. This section compares two approaches to this problem that make different assumptions as to the independence of individual cognitive biases. The approaches reflect a tradeoff between the quality of the selected bias combination and the computing time required to make the selection. The first method looks for individual biases that improve performance of the learning algorithm, and the second evaluates all combinations of available biases.

The first method for selecting biases requires the following steps:

1. Design a baseline representation for the learning problem.
2. Use cross validation to test the effect of incorporating each cognitive bias into the baseline representation independently.
3. Select all biases that improve performance with respect to the best baseline representation, choosing the best-performing implementations and parameters for those biases, and breaking ties randomly.

If we examine the table that summarizes the effect of individual cognitive biases on RP antecedent prediction (Table 8), we see that only the recency bias would be selected in step 3. The subject accessibility and STM biases alone do not improve performance of the learning algorithm with respect to the best baseline representation. In addition, step 3 specifies that the right-to-left recency labeling implementation of the recency bias should be selected because that implementation posted the highest accuracy (81.25% correct). As a result, this method for bias selection chooses only the right-to-left labeling recency bias for the RP antecedent task. Although this method naively assumes that there is no interaction among cognitive biases, it requires only that the effects of individual biases be measured on the training data.

The second method for bias selection exhaustively enumerates all combinations of available cognitive biases and chooses the combination that peforms best in cross-validation testing. This method makes no assumptions about cognitive bias interaction, but requires much more time to select the appropriate bias combination. For the RP antecedent task, this approach to cognitive bias selection requires 240 10-fold cross validation runs on the training data.[9]

We used the exhaustive approach to cognitive bias selection for the RP task and found that one cognitive bias combination involving three biases achieved better results than the best-performing cognitive bias pair. It is shown in Table 13 with the previous best-performing variations. This combination includes the recency bias (recency combination), the STM bias (limit = 5), the missing cues bias (weight = 0.2), and the subject accessibility bias (weight = 2). Results shown in this table clearly indicate that performance of the learning algorithm increases steadily as more biases are added to the baseline representation. Only when all biases are included does the learning algorithm perform significantly better than the hand-coded rules (84.17% correct vs. 80.67% correct) at nearly the 95% confidence level.

[9] There are four baseline variations, four subject accessibility variations, three recency bias variations, and five short term memory variations.

Bias Category	**Bias and Parameters**	**% Correct**
Best Bias Combination	recency combination STM, STM limit = 5 subject accessibility, weight = 2 (missing cues, weight = 0.2)	84.17
Best Pair of Biases	recency combination STM, STM limit = 6 (missing cues, weight = 0.5)	83.33
Best Individual Bias	recency, r-to-l labeling (missing cues, weight = 0.5)	81.25
Best Baseline	(missing cues, weight = 1)	78.75

Table 13: Exhaustive Approach to Cognitive Bias Selection

6 Conclusions

This paper has shown that cognitive processing limitations can serve as a domain-independent source of bias for cognitive learning tasks. We have concentrated on a learning task from natural language processing and explored the effects of four well known cognitive biases on this task: (1) the present vs. missing cues bias, (2) the subject accessibility bias, (3) the recency bias, and (4) short term memory limitations. We have shown that cognitive biases can be automatically and explicitly incorporated into an instance representation. Moreover, our experiments using a nearest-neighbor learning algorithm to determine the antecedents of relative pronouns indicate that the learning algorithm improves as more cognitive biases are incorporated into the instance representation. Similar results were obtained for the same task and a subset of the cognitive biases investigated here [Cardie, 1992b] using the COBWEB conceptual clustering system [Fisher, 1987] instead of the 1-nearest neighbor case-based algorithm. In future work, we plan to (1) examine additional cognitive biases, (2) evaluate the effect of using cognitive biases in new problem domains, (3) incorporate cognitive biases into a variety of learning algorithms, and (4) identify problem-independent cognitive bias interactions so that many cognitive bias combinations could be reflected in an instance representation without requiring cross-validation to test their effectiveness.

References

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

[Ahn and Medin, 1992] Ahn, W. and Medin, D. L. 1992. A Two-Stage Model of Category Construction. *Cognitive Science* 16(1):81–121.

[Bourne *et al.*, 1976] Bourne, L. E.; Ekstrand, B. R.; Lovallo, W. R.; Kellog, R. T.; Hiew, C. C.; and Yaroush, R. A. 1976. Frequency Analysis of Attribute Identification. *Journal of Experimental Psychology* 105:294–312.

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

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

[Cardie, 1992b] Cardie, C. 1992b. Using Cognitive Biases to Guide Feature Set Selection. In *Proceedings of the Fourteenth Annual Conference of the Cognitive Science Society*, Indiana Univeristy, Bloomington, IN. Lawrence Erlbaum Associates. 743–748. Also in Working Notes of the 1992 AAAI Workshop on Constraining Learning with Prior Knowledge, San Jose, CA.

[Cuetos and Mitchell, 1988] Cuetos, F. and 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.

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

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

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

[Fisher, 1987] Fisher, D. 1987. Knowledge Acquisition Via Incremental Conceptual Clustering. *Machine Learning* 2:139–172.

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

[Gernsbacher *et al.*, 1989] Gernsbacher, M. A.; Hargreaves, D. J.; and 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, 1990] Gibson, E. 1990. Recency Preferences and Garden-Path Effects. In *Proceedings of the Twelfth Annual Conference of the Cognitive Science Society*, Massachusetts Institute of Technology, Cambridge, MA. Lawrence Erlbaum Associates.

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

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

[Lebowitz, 1987] Lebowitz, M. 1987. Experiments with Incremental Concept Formation: UNIMEM. *Machine Learning* 2:103–138.

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

[McDonald and MacWhinney, 1991] McDonald, J. and MacWhinney, B. 1991. Levels of Learning: A Comparison of Concept Formation and Language Acquisition. *Journal of Memory and Language* 30(4):407–430.

[Medin *et al.*, 1987] Medin, D. L.; Wattenmaker, W. D.; and Hampson, S. E. 1987. Family Resemblance, Concept Cohesiveness, and Category Construction. *Cognitive Psychology* 19:242–279.

[Michalski and Kodratoff, 1990] Michalski, R. S. and Kodratoff, Y., editors. *Machine Learning: An Artificial Intelligence Approach*, volume 3. Morgan Kaufmann, San Mateo, CA.

[Michalski *et al.*, 1983] Michalski, R. S.; Carbonell, J. G.; and Mitchell, T. M., editors. *Machine Learning: An Artificial Intelligence Approach.* Morgan Kaufmann, San Mateo, CA.

[Michalski *et al.*, 1986] Michalski, R. S.; Carbonell, J. G.; and Mitchell, T. M., editors. *Machine Learning: An Artificial Intelligence Approach*, volume 2. Morgan Kaufmann, Los Altos, CA.

[Miller, 1956] 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.

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

[Nicol, 1988] Nicol, J. 1988. *Coreference Processing During Sentence Comprehension*. Ph.D. Dissertation, Massachusetts Institute of Technology, Cambridge, MA.

[Quinlan, 1986] Quinlan, J. R. 1986. Induction of decision trees. *Machine Learning* 1:81–106.

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

[Sundheim, 1991] Sundheim, B. 1991. Overview of the Third Message Understanding Evaluation and Conference. In *Proceedings of the Third Message Understanding Conference (MUC-3),* San Mateo, CA. Morgan Kaufmann. 3–16.