Integrating case-based learning and cognitive biases for machine learning of natural language

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

Running head: Integrating CBL and cognitive biases

August 9, 1999

Integrating case-based learning and cognitive biases for machine learning of natural language

Abstract

This paper shows that psychological constraints on human information processing can be used effectively to guide feature set selection for case-based learning of linguistic knowledge. Given as input a baseline case representation for a natural language learning task, our algorithm selects the relevant cognitive biases for the task and then automatically modifies the representation in response to those biases by changing, deleting, and weighting features appropriately. We apply the cognitive bias approach to feature set selection to four natural language learning problems and show that performance of the case-based learning algorithm improves significantly when relevant cognitive biases are incorporated into the baseline instance representation. We argue that the cognitive bias approach offers new possibilities for case-based learning of natural language: it simplifies the process of instance representation design and, in theory, obviates the need for separate instance representations for each linguistic knowledge acquisition task. More importantly, the approach offers a mechanism for explicitly combining the frequency information available from corpus-based techniques with cognitively-based preferences employed in traditional linguistic and knowledge-based approaches to natural language processing.

Integrating case-based learning and cognitive biases for machine learning of natural language

1 Introduction

In recent years, statistical and machine learning techniques have been shown to significantly reduce the knowledge-engineering effort for building large-scale natural language processing (NLP) systems: they offer an automatic means for acquiring robust solutions for a host of lexical and structural disambiguation problems. Among the learning algorithms that have been used successfully for natural language learning tasks are case-based learning methods where the natural language system processes a text by retrieving stored examples that describe how similar texts were handled in the past. Case-based approaches have been applied to sub-lexical language acquisition tasks, including stress acquisition (Daelemans et al. 1994) and grapheme-to-phoneme conversion (Bosch and Daelemans 1993). They have been used for lexical tagging and partial parsing: e.g. part-of-speech tagging (Cardie 1993a, Daelemans et al. 1996), word-sense disambiguation (Ng and Lee 1996), semantic class tagging (Cardie 1993a), noun phrase chunking (Argamon et al. 1998, Daelemans et al. 1999), and prepositional phrase attachment (Zavrel et al. 1997). Finally, case-based methods have also been employed in higher-level NLP tasks like context-sensitive parsing (Simmons and Yu 1992), text categorization (Riloff and Lehnert 1994), and concept extraction (Cardie 1993a).

It is well known in the machine learning community, however, that the success of a learning algorithm depends critically on the representation used to describe the training and test instances. Unfortunately, the task of designing an appropriate instance representation — also known as *feature set selection* — can be extraordinarily difficult, time-consuming, and knowledge-intensive (Quinlan 1983). This poses a problem for current statistical and machine learning approaches to natural language understanding where a new instance representation or probabilistic model is typically required for each linguistic task.

Feature selection and feature weighting are especially important in case-based learning. *Case-based learning* algorithms (alternately referred to in the literature as memory-based reasoning (Stanfill and Waltz 1986), instance-based learning (Aha et al. 1991), case-based reasoning (Riesbeck and Schank 1989, Kolodner 1993), and lazy learning (Wettschereck et al. 1997)) solve a problem by first creating a case base of previous problem-solving episodes. In classical case-based learning schemes, each case encodes (1) the context in which the problem occurred as a set of *attribute-value* pairs, and (2) the solution to the problem in terms of a *class value*. Then, when a new problem is encountered, the "most similar" case is retrieved from the case base and its class value is presented as the solution to the novel problem. In its simplest incarnation, case retrieval operates via a nearest-neighbor algorithm that counts the number of overlapping features between a test case and each training case and then retrieves the case with the highest score. In this implementation of case-based learning, it is clear that performance will degrade as more and more irrelevant features populate the instance representation. In addition, some features may be more important than others for a particular learning task, but automatically determining appropriate feature weights is a difficult learning problem in itself.

Previous work in case-based learning of natural language has employed a variety of methods for handling the feature set selection problem. Most often, the similarity measure in the case retrieval algorithm performs feature selection and feature weighting. Ng and Lee (1996), for example, have used similarity measures derived from Stanfill and Waltz's (1986) value-difference metric to assign a different weight to each value of a feature. Weights created by the value-difference metric are based on differences in the distribution of feature values across the class values. In contrast, Daelemans et al. (1994) introduce a global feature weighting scheme that has shown good performance across a number of natural language learning tasks. In short, they calculate the information gain for each feature and use it to weight matches on that feature during case retrieval. This feature weighting scheme is global in that it derives a single weight vector for the task that is used for all retrievals from the case base. In related work, we explore a similar idea for automating feature selection rather than feature weighting (Cardie 1993b). Case-based learning algorithms occasionally employ no feature weighting whatsoever — all features selected for inclusion in the instance representation are assumed to be of equal importance (e.g. Simmons and Yu (1992), Riloff and Lehnert (1994)).

This paper proposes a very different approach to the problem of feature set selection for case-based learning of linguistic knowledge. In particular, our algorithm relies on research in psychology and psycholinguistics that has discovered and examined numerous psychological constraints on human information processing. Initially, we will examine a set of well-known cognitive biases:

- **Subject Accessibility:** The first item mentioned in a sentence has been shown to be of special importance. This bias is an example of a more general **Focus of Attention** bias.
- **Recency:** People consistently show a bias toward the use of temporally recency information in sentence interpretation.
- **Restricted Memory:** Short term memory limitations are known to affect a number of language comprehension skills.
- **Semantic and Syntactic Priming:** People tend to respond more quickly to words that are semantically related to entities currently involved in the interpretation process. Similary, people exhibit a language-processing preference for structures that are syntactically consistent with nearby sentence elements.

In the sections that follow, we describe the biases in more detail and show that psychological constraints such as these can be used effectively to guide feature set selection for case-based learning of linguistic knowledge. Given as input the baseline instance representation for a natural language learning task, our algorithm (1) selects the combination of cognitive biases that are appropriate, and (2) uses them to modify the instance representation. Modifications either directly or indirectly change the feature set used to describe all instances. Direct changes to the representation are made by changing or deleting features; indirect changes modify a weight associated with each feature.

We apply the cognitive bias approach to feature set selection to four natural language learning tasks — relative pronoun disambiguation, and predicting the general semantic class, specific semantic class, and part of speech of unknown words. For each task, we find that the accuracy of the case-based learning algorithm improves significantly when one or more relevant cognitive biases are incorporated into the baseline instance representation. In addition, we believe that the cognitive bias approach to feature set selection offers new possibilities for case-based learning of natural language:

- It provides a natural mechanism for combining the frequency information available from corpus-based techniques with cognitively-based preferences employed in traditional linguistic and knowledge-based approaches to natural language processing.
- The cognitive bias approach to feature set selection simplifies and shortens the process of designing an appropriate instance representation for individual natural language learning tasks. System developers can safely include features for *all* available knowledge sources in the baseline instance representation — the irrelevant features will be discarded automatically and the remaining features weighted appropriately.
- In theory, the cognitive bias approach to feature set selection allows the same underlying instance representation to be used across many linguistic knowledge acquisition tasks.

The remainder of the paper is organized as follows. Section 2 introduces the main natural language learning task, relative pronoun disambiguation. Section 3 summarizes psycholinguistic studies of the effect of five selected cognitive processing biases on language processing. We then describe the basic case-based learning algorithm used in our empirical studies (Section 4). Next, we present the details of the cognitive bias approach to feature set selection, using the relative pronoun disambiguation task as a running example. More specifically, we describe the baseline instance representation in Section 5. Section 6 shows how the baseline representation can be modified automatically in response to three cognitive biases — focus of attention, recency, short term memory limitations; it then evaluates the effect of each bias on relative pronoun resolution. Section 7 proposes and evaluates two related algorithms for cognitive bias selection. We extend the set of available cognitive biases and evaluate the entire approach on three lexical tagging tasks in Section 8. We conclude with a discussion of the general implications of the cognitive bias approach to feature set selection and its connection to existing work in machine learning and NLP (Section 9). Readers are referred to a related paper (Cardie 1999) that discusses the same set of experiments and their implications from a machine learning, rather than an NLP perspective.

2 Relative pronoun resolution

This section introduces the first natural language learning task, relative pronoun disambiguation, as a vehicle for describing the cognitive bias approach to feature set selection. For this problem, the goal of the case-based learning algorithm is to disambiguate wh-words (e.g. who, which, where, that) in sentences like:

> Tony saw *the boy* **who** won the award.
> I didn't know **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. In the first sentence, for example, the system should recognize that 'the boy' is the antecedent of 'who' because 'who' refers to 'the boy'. In the second example, the learning algorithm should determine that there is no explicit antecedent for 'who' — in this case, 'who' is not being used as a relative pronoun.

Finding the antecedents of relative pronouns is an important task for natural language understanding systems and especially for the information extraction task within which our learning algorithm was embedded (Lehnert et al. 1992): the antecedent typically fills a semantic role in the underlying functional structure of its clause as well as the relative clause. The semantic interpretation of the first sentence above, for example, should include the fact that 'the boy' is the object of 'saw'. In addition, it should include the fact that 'the boy' is the implicit actor of 'won' even though 'the boy' does not appear in the embedded clause 'who won the award'. Furthermore, relative pronouns occur frequently in the long, multi-clause sentences of many real-world text collections. We focus more specifically on learning disambiguation heuristics for 'who' because this was the most frequent relative pronoun to appear in the information extraction corpus, occurring in about one out of every ten sentences and at a higher frequency in documents that are actually relevant to the information extraction task.

Although finding relative pronoun antecedents seems a simple enough task, there are many factors that make it difficult:[1]

- **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. 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'.
- **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 antecedent depends on the surrounding context.

[1] Locating the implicit position, or 'gap', 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 the gap-finding problem that is consistent with the work presented here.

- **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 'from' prepositional phrase). The position of the antecedent in each case is different, however.

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.	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 gifts from *the children* **who** attended the party are on the table.

Table 1: Antecedents of 'who'

Despite these ambiguities, we will show that a case-based learning system can learn to locate the antecedent of 'who' given a description of the clause that precedes it. In addition, we will show that cognitive processing preferences known to affect human language processing can also be used effectively as representational biases for the case-based learning algorithm. The next section describes those biases and briefly examines psycholinguistic studies of their effects on language processing in general, and on relative pronoun resolution, in particular.

3 Cognitive biases in language processing

For the most part, sentence processing research in psycholinguistics has focused its study on language constructs that are particularly troublesome for people. Long-distance dependencies (e.g. Fodor (1989)) and garden path sentences (e.g. Crain and Steedman (1985)) represent two such classes of difficult and well-studied linguistic phenomena. The four language tasks selected for study in this paper constitute examples of each. Understanding relative clauses requires resolving long-distance dependencies: the relative pronoun antecedent can be quite distant from both the relative pronoun and the relative clause 'gap' that it fills. Lexical decisions like word-sense and part-of-speech disambiguation, on the other hand, are the source of many garden path sentences. The following sentence, for example, normally causes confusion for the reader at the word 'sank': 'The boat floated down the river sank'. The problem is that in an initial syntactic analysis, people tend to interpret 'floated' as a tensed verb rather than a past participle. The hope of psycholinguists is that studies of how people handle these difficult constructs will uncover general properties of language processors — properties that might not have emerged during the study of simpler structures.

Over the past 25 years or so, many such properties of language processors have emerged and have been shown to affect people's linguistic decisions as they interpret language. The principle of Right Association (Kimball 1973), for example, assumes that in the absence of other information, phrases will be attached to a partial syntactic analysis at the rightmost available point. In contrast, the principle of Minimal Attachment (Frazier and Fodor 1978) states that people prefer to attach phrases in a way that minimizes the complexity of the analysis. Thematic role information (i.e. information associated with a verb sense that defines the allowable participants in the action — the actor, patient, theme, instrument) has also been shown to provide strong constraints for sentence analysis, especially in uncovering long-distance dependencies (Tanenhaus et al. 1989b). Others have found that verb subcategorization and transitivity effects can influence structural attachment decisions by preferring syntactic structures that are consistent with these properties of a verb (Clifton et al. 1984, Nicol 1988, Tanenhaus et al. 1989a). Language processing properties have also been identified for lexical disambiguation tasks. Swinney (1979), for example, found that all meanings of ambiguous words are initially activated when the word is encountered. Others, however, have found exceptions to this multiple-access property by showing that context can sometimes limit activation to only the appropriate meaning (Simpson and Krueger 1991). The relative frequency of competing lexical and structural interpretations is also known to affect their plausibility (MacDonald et al. 1994).

This paper focuses on five cognitive properties or biases in language processing: subject accessibility, recency preferences, short term memory limitations, semantic priming, and syntactic priming. Like the general properties listed above, these biases can potentially affect a language processor's ability to interpret a variety of linguistic constructs.

Recency bias. In processing language, people consistently show a bias towards the use of the most recent information (e.g. Gibson (1990), Nicol (1988), Kimball (1973)). 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 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'.

Other studies, however, have shown that the recency bias interacts with other processing preferences. Gibson and Pearlmutter (1994) describe an earlier study (Gibson et al. 1993) that examined the following construct (where NP refers to noun phrase and PREP refers to preposition): *NP1 PREP1 NP2 PREP2 NP3 RC*. In these constructs, the relative clause (RC) could syntactically attach to any one of NP1, NP2, or NP3. In addition, the relative clause verb agreed in number with only one of the three head nouns:

1. [$_{\text{NP1}}$ the *computer*] [$_{\text{PREP1}}$ near] [$_{\text{NP2}}$ the models] [$_{\text{PREP2}}$ of] [$_{\text{NP3}}$ the buildings] [$_{\text{RC}}$ **that** was destroyed in the fire]

2. [$_{\text{NP1}}$ the computers] [$_{\text{PREP1}}$ near] [$_{\text{NP2}}$ the *model*] [$_{\text{PREP2}}$ of] [$_{\text{NP3}}$ the buildings] [$_{\text{RC}}$ **that** was destroyed in the fire]

3. [$_{\text{NP1}}$ the computers] [$_{\text{PREP1}}$ near] [$_{\text{NP2}}$ the models] [$_{\text{PREP2}}$ of] [$_{\text{NP3}}$ the *building*] [$_{\text{RC}}$ **that** was destroyed in the fire]

The antecedent of 'that' is shown in italics for each example. Their results indicate that subjects find attachment to NP3 easiest to comprehend, followed by NP1, and then NP2. Their explanation for this ordering is that there are two independent preference factors in effect: a very strong recency bias prefers NP3; in addition, a predicate proximity principle (Frazier 1991, Gibson 1991) prefers attachment to NP1 because it is structurally closer to the predicate of the clause that will follow, i.e. it is potentially the subject of that clause.

Subject Accessibility. A number of studies have noted the special importance of the first item mentioned in a sentence, which is very often the subject of the sentence (Gernsbacher and Hargreaves 1988, Gernsbacher et al. 1989, Carreiras et al. 1995). In particular, Gernsbacher et al. (1989) use 'reactivation' techniques to show that the accessibility of the first actor of a sentence remains high even at the end of a sentence. More specifically, a target word is briefly flashed on the computer screen at some point during aural presentation of a sentence. Subjects are asked to decide whether or not the visually presented word was a real word and press the appropriate button. Faster response to a word semantically related to the target word than to an unassociated control is attributed to priming by the target. Gernsbacher et al. find strong reactivation effects for the first actor mentioned in a sentence even when that actor is not the agent of the verb and even when reactivation is tested at the end of the sentence when the distance to the first actor is greatest.

We view the subject accessibility bias as 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.

Restricted Memory Limitations. Psychological studies have determined that people can remember at most seven plus or minus two items at any one time (Miller 1956). This restriction on short term memory capacity has been noted as a cognitive processing constraint that affects many language processing skills. Syntactic structures, for example, are believed by many to be constructed via the application of a grammar, which is accessed as needed rather than stored in its entirety in memory. Thus, it is assumed that the computation involved in syntactic analysis places demands on working memory that allow the parser to pursue only a single analysis at a time (Frazier 1979).

More recently, Daneman and Carpenter (1980, 1983) have studied the relationship between working memory capacity and language processing. They first estimate a subject's working memory capacity using a Reading Span Test.[2] They next measure a subject's ability to find the referents of pronouns over varying distances and find that working memory capacity accurately predicts the differences observed in their subjects: Subjects with a larger working memory capacity can identify pronoun references over longer distances. They also find that working memory capacity is correlated with a subject's ability to integrate semantic information within and across sentences.

Furthermore, 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. Readers with low working memory capacity showed poorer comprehension of relative clauses with object gaps than high capacity readers, this in spite of spending more time reading the critical portions of test sentences. This same study determined, however, that pragmatic cues provided via an appropriate context can improve performance of both types of readers. In addition, they 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.

Finally, 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.

Semantic and Syntactic 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. Detecting semantic priming effects is a primary means for supporting many cognitively-based hypotheses for on-line interpretation tasks (Fodor 1989, Swinney et al. 1988), including those in a number of the studies mentioned above (e.g. Gernsbacher et al. (1988, 1989), Nicol (1988)). Methods that detect semantic priming effects provide a window into the workings of semantic interpretation by indicating the time course with which each entity in a sentence is incorporated into the meaning representation for that sentence.

Analogously, we will later refer to a syntactic 'priming' bias. This bias is meant to combine into a single bias several syntactic effects. Namely, it represents the language processor's preference for structures that are consistent with syntactic properties of nearby sentence elements, e.g. subcategorization properties, transitivity, person, number, and gender.

Note that existing systems for natural language understanding rely heavily on cognitive biases, incorporating them as heuristics or constraints at many levels of text analysis. Case-based learning approaches also often implicitly make use of one or more cognitive biases. For each natural language learning task, for example, an appropriate context window must be selected for inclusion in the case representation. Nevertheless, for both traditional and corpus-based NLP systems, it is difficult to determine the relative importance of individual biases for a specific language understanding problem. In addition, it can be difficult to specify complex bias interactions algorithmically. Furthermore, as the linguistic knowledge sources made available to the system change over time, it becomes increasingly difficult to modify and maintain an NLP system that encodes processing biases and linguistic knowledge in the same monolithic data structures.

Our approach makes explicit the use of cognitive biases. In addition, it separates the representation of linguistic knowledge and the implementation of cognitive processing preferences. In the sections below, we show first how cognitive biases can be incorporated automatically into a case-based approach to natural

[2] The Reading Span Test determines the length of the longest sequence of sentences for which subjects can remember the final word.

language processing. We also show how multiple biases can be combined. Finally, we present a method for learning the relative strength of each bias for particular natural language tasks.

4 The case-based learning algorithm

Throughout the paper, we employ a simple k-nearest neighbor case-based learning algorithm. The goal of the training phase is to collect a set of cases that describe ambiguity resolution episodes for a particular problem in text analysis. Each case is a set of features, or attribute-value pairs, that encode the context in which the ambiguity occurred. In addition, each case is annotated with a class value that describes how the ambiguity was resolved in the current example. For part-of-speech tagging, for example, the class value is the syntactic category associated with the current word; for semantic feature assignment, the class value is the semantic class associated with the current word; for structural attachment decisions, the class information indicates the position of the preferred attachment point.

After training, the case base can be used to resolve ambiguities in novel sentences. Given a test case that describes the context of an ambiguity encountered in a novel sentence, the case retrieval algorithm compares the problem case to those stored in the case base, finds the most similar training case, and then uses the class information to resolve the current ambiguity. Unless otherwise specified, we employ the following weighted 1-nearest-neighbor (1-nn) case retrieval algorithm in all of our experiments:

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 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.

2. Return the case with the highest score along with any ties.

3. If a single case is retrieved, return its class value as the 'solution' for the test case. Otherwise, let the retrieved cases vote on the class value.

5 Relative pronoun resolution using the baseline instance representation

This section describes the baseline instance representation for the relative pronoun resolution problem. It is considered a 'baseline' in that it explicitly encodes none of the cognitive biases that will be examined in subsequent sections. Throughout the paper, we assume that the learning algorithm is embedded in a larger NLP system that processes input sentences and generates training and test cases from them. This has important ramifications for the case representation, which can include attribute-value pairs for only those linguistic constructs and knowledge sources that are available to the NLP system. In our work, the larger NLP system is the CIRCUS information extraction (IE) system (Lehnert 1990, Lehnert et al. 1991, Lehnert et al. 1992, Lehnert et al. 1993). In general, an information extraction system takes as input a set of unrestricted texts and 'summarizes' each 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 (Lehnert and Sundheim 1991, Chinchor et al. 1993, Cardie 1997).

5.1 The baseline instance representation

For relative pronoun resolution, the IE system generates one instance for every occurrence of 'who' that appears in training or test texts. Each training instance 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 single class value that describes the position of the correct antecedent for 'who' in each example. The details of the baseline instance representation depend entirely on the NLP system used to generate the cases — the case should encode the state of the system at the point of relative pronoun disambiguation. Key features of our IE sentence analyzer that influence the case representation include the following:

- The IE system 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 or appositives, or perform prepositional phrase attachment.
- The IE system uses one or more semantic features to describe every noun and adjective in its lexicon. For example, 'mayor' is tagged as *human*; 'ELN' is tagged as an *organization* and the noun 'murder' is tagged as an *attack*.
- The IE system keeps track of the most recently recognized linguistic entity, e.g. word, phrase, punctuation mark.

In addition, the baseline instance representation should include information for each element of the wh-word context. Because the antecedent of 'who' usually appears as one or more phrases in the preceding clause, training and test 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 2, in which there are two phrases in the clause preceding 'who' — 'the man' and 'from Oklahoma'.

1. For each phrase, there is one CONSTITUENT feature that denotes the syntactic class and position of the phrase as it was encountered by the IE sentence analyzer. 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.
2. In addition, there is a SEM feature for each phrase that provides its semantic classification if one is available from the system lexicon.[3] The SEM feature for the subject (S-SEM *human*) indicates that 'the man' is *human*; 'from Oklahoma' specifies a *location*.
3. 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'.
4. 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.
5. 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 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 3, for example, the direct object of 'thank' is the conjunction 'Nike and Reebok'. In the IE system, and therefore in our instance representation, however, '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 the IE system, there is no SEM feature for verbs.

Every training instance is also annotated with class information — a list of the CONSTITUENT attributes that represent the position of the antecedent of 'who' for the current example or *none* if there is no antecedent. In the sentence of table 2, for example, the antecedent of 'who' is 'the man'. Because this phrase

[3] Noun phrases in the information extraction system are described by one of eleven semantic features: human, proper-name, location, entity, physical-target, organization, weapon, attack, pro (pronoun), time, and quantity. A subset of these features are specific to the terrorism information extraction domain from which the training instances were extracted. A different set would be required for texts from a different domain.

Table 2: 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)	

Table 3: Baseline instance representation for 'I thank Nike and Reebok, who ...'

Phrases	Features	
I	(S	*t*)
	(S-SEM	*pro*)
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)	

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 3, 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. For example, 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). For appositive and predicate nominative constructs, exact matches on any of the antecedent options is considered correct.

It should be noted that training instances are generated by the IE system automatically from unrestricted texts as a side effect of sentence analysis. Only the correct antecedent must be specified by a human supervisor.

5.2 The relative pronoun data set

The relative pronoun data set contains 241 instances — all examples of 'who' from 150 texts from the MUC-3 terrorism corpus. This corpus was developed in conjunction with the third and fourth Message Understanding Conferences (MUC-3 1991, MUC-4 1992), which are performance evaluations of state-of-the-art information extraction systems. The collection contains 1300 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. Documents vary in length from a paragraph to a page or two. Average sentence length in the corpus is 27 words; the average length of each article is 12 sentences.

In the baseline experiments, all weights for the 1-nn classifier are set to one. As noted above, the system uses the antecedent of the retrieved case(s) to predict the antecedent of 'who' in novel sentences: 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 performance of the learning algorithm for relative pronoun resolution depends, in part, on task-independent characteristics of the data set. First, instances contain between two and 31 features and represent clauses that have from one to 11 phrases. 186 of the cases (77%) are unique. In addition, the antecedent class takes on 60 distinct values across the data set. In particular, there are 10 instances (4%) for which no other instance in the data set has the same antecedent. As a result, the case-based algorithm has no hope of getting these right when using the baseline instance representation.

Because the data set is created automatically, it can contain errors. For the experiments in this paper, however, we made one pass through the data set to correct a small number of obvious parsing and semantic class disambiguation errors. The data set does, however, contain 'ambiguous' instances — instances that have the same set of attribute-value pairs, but a different antecedent value. There are four such pairs. Three of the four have one instance that represents a predicate nominative construction, e.g. '*d'Aubuisson* is *another assassin* **who** ...' In these situations there are three options for the antecedent — the subject, the object, or both the subject and the object. The 'conflicting' instance in each case has only the object as an allowable antecedent, e.g. 'The army leaders threatened *people* **who** ...' This type of conflict does not preclude obtaining a correct answer since the intersection of the antecedents for the conflicting pair of instances is non-empty. The final pair of ambiguous instances, however, represents an actual conflict:

> The invasions will punish *someone* **who** ...
> The investigations will undoubtedly show us **who** ...

In the first example, 'who' refers to the direct object; in the second, there is no antecedent. Our instance representation is too coarse to differentiate the two contexts for each use of 'who'. Including additional lexical features to distinguish 'someone' and 'us' as different types of pronouns would be one way to remove the conflict.

The difficulty of the relative pronoun disambiguation task depends on at least three additional factors — the complexity of the antecedent, its distance from the relative pronoun, and the level of potential ambiguity in determining the antecedent. In our data set, 16% of the instances involve antecedents that are conjunctions or appositives or both. These are more complex than single-phrase antecedents and should be harder for the learning algorithm to locate. Also difficult are instances with large distances between 'who' and its antecedent. The maximum such distance for our data set is seven phrases. However, the average such distance is 1.6 phrases, which indicates that biases towards choosing recent phrases as the antecedent should work well. Unfortunately, 51% of the cases in the data set involve legitimate syntactic ambiguity with respect to relative pronoun resolution. These are sentences like 'I saw $[_{\mathrm{phrase1}}$ the daughter] $[_{\mathrm{phrase2}}$ of the colonel] **who**...' or 'I walked $[_{\mathrm{phrase1}}$ with the man] $[_{\mathrm{phrase2}}$ from Detroit] **who**...' Syntactically, the antecedent can be either the head noun of phrase_1 or phrase_2. The case-based learning algorithm will have to use semantic class information or punctuation to correctly handle many of these cases.

5.3 Evaluation using the baseline instance representation

The results for the case-based learning algorithm using the baseline case representation are shown in table 4. The first row shows the accuracy of the 1-nn algorithm using leave-one-out cross validation. The remaining rows show the performance of five additional baseline systems. The first 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 legitimate antecedents of 'who' are characterized by semantic features other than *human*. Unfortunately, requiring more complicated semantic feature combinations like those of the third rule (Most Recent Human or Proper Name) performs worse than the Most Recent Constituent default rule. Note that none of the default rules can locate multi-phrase antecedents.

Table 4: Results for the baseline representation, default rules, hand-coded heuristics, and the IG-CBL feature-weighting algorithm (% 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

The fourth baseline system of table 4 employs a set of hand-crafted heuristics for relative pronoun resolution that were developed for use with the IE system in the MUC-3 performance evaluation. The heuristics consisted of approximately 30 rules. These rules had access to 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 final row in the table shows the performance of IG-CBL (Cardie and Howe 1997), a weighted k-nearest neighbor algorithm that automatically derives feature weights and that has been applied successfully to a number of natural language learning problems including part-of-speech tagging, semantic class tagging, and prepositional phrase attachment (Cardie 1993b, Daelemans et al. 1996). IG-CBL is a straightforward composition of two existing methods. It first uses a decision tree for feature selection as described in Cardie (1993b). The goal in this step is to prune irrelevant features from the representation so that the CBL algorithm can ignore them entirely. IG-CBL then assigns each remaining feature a weight based on the feature's information gain as measured across the training cases as proposed in van den Bosch and Daelemans (1993) and Daelemans et al. (1997). The intent here is to weight each feature relative to its overall importance in the data set. 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.

In summary, the results in table 4 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 variations are indistinguishable from one another in terms of statistical significance. In addition, all three significantly outperform the IG-CBL feature weighting method. We believe that IG-CBL works so poorly on this natural language learning data set 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. Finally, 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, for subject antecedents, and for detecting when 'who' is not being used as a relative pronoun (antecedent = *none*).

6 Incorporating the cognitive biases

In the following subsections, we modify the baseline case 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. Modifications to the instance representation occur in one of two forms: Direct changes to the representation modify or remove attributes from the relative pronoun feature set; indirect changes modify the feature weight vector. Unless otherwise stated, all results are leave-one-out cross validation averages and all statements of statistical significance are at or above the 95% confidence level ($p < 0.05$). We use both χ^2 and Mc Nemar's significance tests.[4]

6.1 Incorporating the subject accessibility bias

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 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 5: Results for the subject accessibility bias (% correct)

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

Table 5 shows the effects on relative pronoun resolution when the subject weight is 2, 5, 8, and 12. The results indicate that 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. Mc Nemar's test indicates significant differences between the baseline representation and the subject-weighted representation when the weight is five or 12. Higher subject weights were tested, but provide no improvement in performance.

At first these results may seem surprising; however, the baseline representation produced by the IE 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 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

[4]The χ^2 test indicates whether or not changes in classifier accuracy are statistically significant based on the frequencies that underlie the average accuracies. Mc Nemar's test also takes into account the relative performance on individual instances in the test set. Two classifiers with the same accuracy can exhibit a significant difference in performance using Mc Nemar's test if they misclassify very different subsets of instances. See Dietterich (1998) for a description of Mc Nemar's and other significance tests.

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.

6.2 Incorporating the recency bias

Next, we translate the recency bias into representational changes for the training and test instances in two ways. In the first approach, the features are renamed according to the position of 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 (l-to-r) labeling that is more natural for the parser and that is espoused in the baseline instance representation. Table 6 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 of the figure indicate the original baseline and r-to-l recency representations, respectively. 'In Congress', for example, assumes the attribute PP-1 in the r-to-l labeling 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 right-to-left ordering yields a different feature set and, hence, a different instance representation. Intuitively, the modified 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 (l-to-r) representation, on the other hand, labels the most recent constituents and the antecedents for each instance with different attributes, making it unlikely for one case to retrieve the other.

Table 6: 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 *prep-phrase*)	who ...	(PREV-TYPE *prep-phrase*)
(ANTECEDENT *(DO)*)		(ANTECEDENT *(NP-2)*)

In the second approach to incorporating the recency bias, we increment the original weight associated with each feature as a function of its proximity to the wh-word (see table 7). To create a recency-biased weight vector, the feature associated with the element closest to the relative pronoun receives a weight of 10 (other values will be tested in subsequent sections), and the weights are decreased by one for each of the preceding features until reaching zero. This implementation of recency weighting is depicted in the Recency Weight column of table 7. Note that the PREP, SEM, and MARK features for each phrase obtain a weight that depends on when the feature is recognized by the parser in its left-to-right processing of the text:

the preposition, if any, is identified first; then semantic class information; and finally the punctuation mark or conjunction that follows the phrase. Recency weights are then added to the original 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.

Table 7: Incorporating the recency weights (with a maximum weight = 10)

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...				

The results of experiments that use the recency representations separately and in a combined form are shown in table 8. To combine the r-to-l labeling and recency weighting biases, the system first relabels the attributes of an instance using the right-to-left labeling and then initializes the weight vector using the recency weighting procedure described above. The table shows that recency weighting and r-to-l labeling have relatively little effect on wh-word antecedent prediction when applied in isolation: The increases in performance for these runs 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). Mc Nemar's test also indicates significant differences with respect to the r-to-l labeling representation.

We believe that the combined recency bias (R-to-L+RecWt) 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.

Table 8: Results for the recency bias representations (% 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**

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 5.2, the data set encoded using the baseline case representation exhibits 60 unique antecedents. After incorporating the r-to-l recency bias, this number is reduced to 39. The number of instances for which no other instance in the data set has the same antecedent is similarly reduced — from 10 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 are unique vs. 186 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 10 times. In general, the combined recency representation does markedly better than the baseline in recognizing when 'who' is not being used as a relative pronoun. Although the combined recency 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. Five 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.)

> Since it is quite certain that Vargas Llosa, the presidential candidate of the Democratic Front, will win the first round, the question$_{baseline}$ now is **who** will be his rival... (Correct antecedent is none$_{recency}$.)
>
> Ordonez added: I was aware that Pena wanted to get rid of somebody, but I$_{baseline}$ never learned **who** they were going to kill until... (Correct antecedent is none$_{recency}$).
>
> All the democrats$_{recency}$ of the Americas$_{baseline}$ **who** see...
>
> He put himself at the disposal$_{baseline}$ of the Spanish residents$_{recency}$ in El Salvador **who** want to leave the country...
>
> Spaniard Jose Maria Martinez$_{recency+baseline}$, Frenchman Roberto Lisandy$_{recency}$, and Italian Dino Rossy$_{recency}$, **who** were staying...

6.3 Incorporating the restricted memory bias

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

To more rigorously apply this *restricted memory* bias (RM) to the baseline representation, we let n represent the memory limit. Then, for each test case, the learning algorithm selects the n features with the highest weights, choosing randomly in case of ties; it then sets the weights for the 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.[5]

Table 9: Results for the restricted memory bias representation (% correct).

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

Results for the restricted memory bias representation with n set to one of 3, 5, 8, and 12 are shown in table 9. (Because of the random component of the RM bias, results are averages across five separate 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

[5] Another implementation of the restricted memory bias might select all features associated with n phrases. Since other biases can modify weights for individual features rather than for entire constituents, this alternative implementation would be more difficult to combine with other biases.

$n = 3$. (Mc Nemar'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.

The effect of the restricted memory bias depends on the number of features in the test case and the memory limit. In particular, the nearest-neighbor algorithm bases its computations on the features of the test case rather than on a feature set that is normalized with respect to all features occurring in the training set. For the relative pronoun data set, 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. Representations that use smaller memory limits tend to perform poorly with respect to the baseline representation on instances with many features.

6.4 Discussion of results

Table 10 provides a summary of the best-performing variation of each bias implementation as well as the best baseline systems. The table shows that, individually, none of the cognitive bias implementations improves the accuracy of relative pronoun antecedent prediction over the baseline representation. Increases in performance begin to appear when applying the recency weighting and r-to-l labeling in isolation; however, these increases are not statistically significant. Neither the subject accessibility nor the restriced memory biases is able to boost performance in isolation. 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 when more than one bias is applied to the baseline instance representation.

Table 10: Individual cognitive bias summary. 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**

In summary, this section presented and evaluated basic implementations for three cognitive biases. Although designed for use with the sentence analyzer of our information extraction system, these implementations are fairly general and should be applicable across a wide selection of NLP systems: Table 11 summarizes the assumptions and parameters for each bias implementation. One can also redefine the bias implementations to better match the characteristics of the NLP system that generates training and test cases. After specifying bias implementations, the cognitive bias approach to feature set selection can be applied to a variety of natural language learning tasks. For each problem, the user need only specify the bias to apply and any associated parameters. Upon invocation of the case-based learning algorithm, modifications to the baseline instance representation in response to the selected cognitive bias are then performed completely automatically.

Table 11: Cognitive bias modifications

Bias	Assumptions	Parameters
Focus of Attention (subject accessibility)	–	Constituent associated with object of focus (e.g. the subject); Weight increment.
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 can be ordered according to recency w.r.t. some fixed point.	Maximum recency weight.
Restricted Memory	–	Size of memory.

7 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: when both implementations of the recency bias were instantiated simultaneously, the accuracy of the learning algorithm surpasses that of the baseline representation and the hand-coded heuristics. Furthermore, the same experiments indicate that cognitive bias interactions may play an important role in selecting which biases to apply to a particular learning task. For example, the combined recency representation performed quite well in spite of the small gains produced by the recency weighting in isolation. Interactions occur for other bias combinations as well. When adding the subject accessibility bias to the combined recency representation, the performance of the resulting representation varies depending on the weight associated with the subject feature: There is an insignificant increase in performance with respect to the combined recency representation when the subject weight is two; but a statistically significant decrease in performance when the subject weight is 12. Similarly mixed results occur when simultaneously applying the restricted memory and subject accessibility biases under varying memory limitations and subject weight settings.

It is clear then, that selecting the combination of cognitive biases that will achieve the best performance for a particular natural language learning task can be difficult. The problem is exacerbated by the fact that our computational model is an extremely impoverished model of human language processing: The feature vector representation at best only coarsely encodes the information sources available to humans at a linguistic decision point. The bias implementations are also necessarily simplified with respect to cognitive biases in people. As a result, even cognitive preferences that are known to be important for particular human linguistic tasks may not improve performance in the case-based learning system; and these biases should be discarded from consideration. 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.

7.1 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 1. It is the inner cross validation (step 1b) that selects the biases; the outer cross validation is the standard evaluation methodology. The primary difference between the greedy and exhaustive 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. The subsections that follow describe and evaluate each bias selection approach.

---

1. Divide the data into n partitions. For each partition,
   (a) Call that partition the *test partition* and all remaining data the *training data*.
   (b) Divide the training data into m partitions.
      i. For each partition,
         A. Call that partition the *selection data*.
         B. Call the remaining data from the training set the *learning data*.
         C. **Apply each cognitive bias combination (and associated parameter settings) to the learning data in turn. Test on the selection data.**
      ii. For each cognitive bias combination tested, take the average of the accuracies across the selection data partitions.
      iii. Select the cognitive bias combination with the highest accuracy.
   (c) Apply the selected bias(es) on the training data and test on the test partition.
2. Return the average of the accuracies for each test partition.

Figure 1: The cross validation algorithm as used by both cognitive bias selection algorithms. The greedy and exhaustive bias selection algorithms differ only in the method each uses to instantiate the step in boldface.

7.1.1 Greedy bias selection

In general, the greedy bias selection algorithm operates by incorporating the best of the remaining *individual* biases, one at a time, while accuracy on the selection data improves or remains constant. More specifically, the greedy approach to bias selection iterates over step 1b, adding one more bias with each iteration. The bias combinations under consideration in each iteration are determined in the boldface step of figure 1. To start, this set is initialized to include all individual biases under a variety of parameter settings: e.g.

{ r-to-l labeling,
recency weighting with max wt = 10, recency weighting with max wt = 20, . . . ,
combined recency with max wt = 10, combined recency with max wt = 20, . . . ,
subject accessibility with wt = 2, subject accessibility with wt = 3, . . . ,
restricted memory with limit = 3, restricted memory with limit = 4, . . . }.

Each bias in the set is applied to the learning data and tested on the selection data. The bias with the highest accuracy is added to the set of selected biases. In the next iteration, each of the remaining biases[6] is applied to the learning data, in turn, in combination with the previously selected bias. The best-performing pair then becomes the set of selected biases. This procedure continues while accuracy on the selection data is non-decreasing.

7.1.2 Exhaustive bias selection

The exhaustive bias selection algorithm operates simply by testing all combinations of the 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 1) as follows:

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

[6] After each iteration, alternative variations of the newly selected bias are removed from the set of available biases. If, for example, the recency-weighting representation with max wt = 10 were the first bias selected, then all recency biases are deleted from consideration for the next round of bias selection.

- For every bias combination in L,
   1. Apply the bias combination to the learning data.
   2. Test on the selection data.

In comparison to greedy bias selection, exhaustive bias selection makes few assumptions about cognitive bias interaction; on the other hand, the method is much more expensive in terms of computing time. 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.

7.1.3 Merging bias representations

Both the greedy and exhaustive bias selection algorithms require a systematic procedure for merging the representations produced by two or more individual biases. This is accomplished in the following manner:

1. Incorporate any bias that relabels attributes (e.g. r-to-l labeling).
2. Incorporate biases that modify feature weights by adding the weight vectors proposed by each bias to the original weight vector (e.g. recency weighting, subject accessibility).
3. Incorporate biases that discard features (e.g. restricted memory 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.

7.2 Results on relative pronoun disambiguation

The sections below report the results for both greedy and exhaustive bias selection on the relative pronoun data set.

7.2.1 Greedy bias selection

Table 12 shows the results of applying the greedy bias selection algorithm on the relative pronoun disambiguation task. For these runs, both levels of cross validation used 10 partitions ($n = 10$, $m = 10$). The table indicates overall accuracy and lists the selected biases and the number of partitions for which the combination was selected. One can see that there is 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 10 partitions with a subject weight of three in all cases. The overall 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 (at the 0.05 level or better for both significance tests).[7]

7.2.2 Exhaustive bias selection

Table 13 shows the results of applying the exhaustive bias selection algorithm to the relative pronoun task. Again, we see that there is a fairly uniform selection of cognitive biases across partitions: all three biases are always selected. In particular, exhaustive bias selection chooses the combined recency representation with a maximum weight of 10; 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 overall accuracy of the representations created using exhaustive bias selection is 89.6%. Like greedy bias selection, exhaustive

[7]Note that 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.

Table 12: Greedy bias selection results. The table shows the selected biases and the number of 10-fold cross validation partitions for which the bias was selected. Accuracy refers to % correct.

Selected Biases			Number of Partitions
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

Biases tested:
- R-to-L (Right-to-Left Recency Labeling)
- RecWt (Recency Weighting): max wt = {10, 20}
- Rec Combo (Combined R-to-L and RecWt): max wt = {10, 20}
- Subj (Subject Accessibility): wt = {2, 3, 4, 6, 7, 10, 12}
- RM (Restricted Memory): limit = {3, 4, 5, 6, 7, 8, 10, 12, 15, 20, 25, 30}

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 at the 0.05 level or better using both significance tests. Significance tests indicate no difference in performance when compared to the greedy selection algorithm.

Table 13: Exhaustive bias selection results. The table shows the selected biases and the number of 10-fold cross validation partitions for which the bias was selected. Accuracy refers to % correct.

Selected Biases			Number of Partitions
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:
- R-to-L (Right-to-Left Recency Labeling)
- RecWt (Recency Weighting): max wt = {10, 20}
- Rec Combo (Combined R-to-L and RecWt): max wt = {10, 20}
- Subj (Subject Accessibility): wt = {2, 3, 4, 6, 7, 10, 12}
- RM (Restricted Memory): limit = {3, 4, 5, 6, 7, 8, 10, 12, 15, 20, 25, 30}

7.2.3 Discussion and summary of results

Table 14 summarizes the results of applying the cognitive bias approach to feature set selection to the relative pronoun task. The results are divided into three categories — baseline results, results for the best individual bias implementations, and results after incorporating additional biases using both bias selection algorithms. The table clearly shows that performance of the learning algorithm increases steadily as relevant biases are added to the baseline representation. As noted earlier (Section 6.4), 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: It is only when more than one cognitive bias is incorporated that the resulting case-based learning system performs significantly better than the hand-coded heuristics.

Table 14: Bias selection summary. Results in boldface indicate significance with respect to the baseline representation, the best default rule, and the hand-coded heuristics. The * indicates significant improvements with respect to the combined recency representation.

Baselines	Best Default Rule	77.6
	Baseline Representation	78.8
	Hand-Coded Heuristics	80.5
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 (R-to-L labeling, Rec Wt w/max = 10, RM limit = 5–7, Subj wt = 3)	**89.2***
	Exhaustive Bias Selection (R-to-L labeling, Rec Wt = 10, RM limit = 5 or 8, Subj wt = 2)	**89.6***

A closer look at the results reveals that the increase in performance in response to incorporating the subject accessibility and restricted memory biases together with the combined recency representation is nearly monotonic — the 'all-biases' version misses just one instance that the combined recency representation gets correct:

> I have deep respect for the officers$_{recency}$ in the field$_{all-biases}$, **who** fight every day.

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 constrast, the average 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, syntactically ambiguous relative pronoun constructs (Section 5.2) are responsible for 73% of the differences between the all-biases versions and the combined recency representation. Three such examples follow:

> The government publicly shows the horror$_{recency}$ of women$_{all-biases}$ **who** have been raped in the prisons...

> According to his statements given to DAS last March, not only did Israeli mercenaries train assassins in middle Magdalena, there was also a group$_{all-biases}$ of British mercenaries$_{recency}$ **who** gave lessons to 50 students.

> 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 last two examples show that these syntactically difficult cases are semantically difficult as well: it is often hard for a person to provide consistent antecedent information in the presence of collective or mass nouns. 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, 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 should 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.

8 Lexical tagging tasks

The last two sections showed that the cognitive bias approach to feature set selection can be used to significantly improve performance of case-based learning of relative pronoun disambiguation heuristics. The goal of this section is to demonstrate the generality of the approach by (1) applying the method to three additional natural language learning tasks; and (2) expanding the set of available cognitive biases. Both extensions will serve to further test the robustness of the cognitive bias selection algorithm.

8.1 Handling unknown words

The three new natural language learning tasks address problems encountered by an NLP system when it reaches unknown words. 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. Thus, all three tasks are lexical tagging tasks. Assume, for example, that the word 'general' was an unknown word and that the NLP system receives as input 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 case-based learning algorithm should infer that the part of speech of 'general' is an adjective; in the second sentence, however, 'general' is a noun modifier. Similarly, the case-based learning algorithm should indicate that 'general' is being used in its 'universal entity' sense in the first case (with no secondary semantic feature), but as a 'person:military officer' in the second.

Our lexical tagging data sets simulate such disambiguation of occasional unknown words: We assume that each content word in the training sentences, in turn, is unknown to the system, but that information on the surrounding words in the sentence is available. More specifically, each lexical tagging data set contains 2056 cases. Like the relative pronoun data set, instances are created automatically by the IE sentence analyzer. Each represents the context in which the NLP 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). As shown in the example of figure 2, each case has 34 features. Twenty of these describe the local context in which the test word occurred: these are the part of speech (P-O-S), semantic class (GEN-SEM, SPEC-SEM), information extraction concept (IE-CONCEPT), and actual lexical item (WORD) corresponding to the two words that precede and follow 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. Each case also includes morphological information for the unknown word (MORPHOL). The remaining 13 features encode the semantic features and information extraction concepts for the major 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. As was the case for relative pronoun resolution, the case representation reflects the syntactic and semantic information available to the NLP system as it processes a text. Note also that 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

Phrases	Features	
Toyota Motor Corp.	(SUBJECT-GEN-SEM	*joint-venture-entity*)
	(SUBJECT-SPEC-SEM	*company-name*)
	(SUBJECT-IE-CONCEPT	*nil*)
has set up	(VERB-GEN-SEM	*nil*)
	(VERB-SPEC-SEM	*nil*)
	(VERB-IE-CONCEPT	*nil*)
	(LAST-PHRASE-GEN-SEM	*nil*)
	(LAST-PHRASE-SPEC-SEM	*nil*)
	(LAST-PHRASE-IE-CONCEPT	*nil*)
	(LAST-PHRASE-SYN-TYPE	*verb*)
(no direct object)	(DO-GEN-SEM	*nil*)
	(DO-SPEC-SEM	*nil*)
	(DO-IE-CONCEPT	*nil*)
a	(PREV2-WORD	*a*)
	(PREV2-P-O-S	*determiner*)
	(PREV2-GEN-SEM	*nil*)
	(PREV2-SPEC-SEM	*nil*)
	(PREV2-IE-CONCEPT	*nil*)
joint	(PREV1-WORD	*joint*)
	(PREV1-P-O-S	*noun-modifier*)
	(PREV1-GEN-SEM	*entity*)
	(PREV1-SPEC-SEM	*nil*)
	(PREV1-IE-CONCEPT	*nil*)
venture	(MORPHOL	*nil*)
firm	(FOL1-WORD	*firm*)
	(FOL1-P-O-S	*noun*)
	(FOL1-GEN-SEM	*joint-venture-entity*)
	(FOL1-SPEC-SEM	*nil*)
	(FOL1-IE-CONCEPT	*nil*)
with	(FOL2-WORD	*with*)
	(FOL2-P-O-S	*preposition*)
	(FOL2-GEN-SEM	*nil*)
	(FOL2-SPEC-SEM	*nil*)
	(FOL2-IE-CONCEPT	*nil*)
Class values:		
part-of-speech	NOUN-MODIFIER	
gen-sem	ENTITY	
spec-sem	NIL	

Figure 2: Baseline instance representation for *'venture'* in 'Toyota Motor Corp. has set up a joint venture firm with Yokogawa Electric Corp. ...'

than a structural attachment decision. However, since two-thirds of the features in the lexical tagging data sets now represent neighboring tokens rather than neighboring phrases, the recency bias may have a different effect. Second, we have already discarded many irrelevant features from the representation. For example, the NLP system could 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. Since the IE sentence analyzer distinguishes lexical decisions from structural attachment decisions, we did not modify the system to include the additional features. The result of this preprocessing step will be an inflated performance for the baseline case representation. In addition, there may be less of a need for biases that discard features, like the restricted memory bias. The new natural language learning problems, 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.

8.2 Incorporating semantic and syntactic priming

We implement two additional biases for use with the lexical tagging tasks: semantic and syntactic priming. First, semantic priming increases the weights for all semantic features in the baseline representation (i.e. the GEN-SEM, SPEC-SEM, CONCEPT, and MORPHOL features) by some specified value. This is clearly a very coarse implementation of this bias: It encourages the case retrieval algorithm to match on semantic features, but ignores the problem of determining which entities' semantic features are most relevant. For this, we will rely on the other cognitive biases. Analogously, we implement syntactic priming by increasing the weights for all features associated primarily with syntactic issues (i.e. the P-O-S, SYN-TYPE, and MORPHOL features) by some specified value.

8.3 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 at the word-level, 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). All experiments employ 10-fold cross validation.

8.3.1 Effect of individual biases

Results for the lexical tagging tasks using each of the cognitive biases in isolation are summarized in table 15. For each bias, the table shows results for only the best parameter setting for that bias.[8] 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 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. Also, the IG-CBL feature-weighting approach works very well although only for part-of-speech tagging does it significantly outperform the baseline representation.

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

[8]The value of k (k-nn) for each NLP problem was chosen via cross validation: part-of-speech prediction, $k = 1$; general semantic class, $k = 5$; specific semantic class, $k = 10$.

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. One might have expected the semantic bias to have a positive effect on the semantic tagging tasks. In terms of information gain, however, the preceding and following parts of speech are nearly as important for semantic tagging as the corresponding semantic features. This is true for general semantic class prediction and, to a slightly lesser degree, for specific semantic class prediction (Cardie 1994). There is, in fact, a direct relationship between parts of speech and semantic classes in that a different part of the semantic class hierarchy corresponds to each part of speech.

In general, the individual bias results 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 (e.g. hidden Markov models and transformation-based learning) that classify tokens based only on information associated with one or two of the immediately preceding tokens.

Table 15: 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 Default Rule Tested	Part of Speech (p-o-s)	General Semantic Class (gen-sem)	Specific Semantic Class (spec-sem)
Default Heuristic (Most Frequent)	81.5	25.6	58.1
Default Heuristic (Weighted Random)	34.3	17.0	37.3
IG-CBL	**90.3**±3.3	64.7±5.7	73.4±3.2
Baseline Representation	89.0±3.7	63.9±5.4	74.8±5.3
Recency Weighting	**92.9**±3.4 (max=11)	**70.4**±3.8 (max=11)	**77.5**±4.0 (max=11)
Semantic Priming	85.7±4.2 (wt=2)	62.0±5.3 (wt=2)	74.5±5.1 (wt=2)
Syntactic Priming	**90.7**±4.1 (wt=6)	61.8±4.4 (wt=2)	73.0±4.9 (wt=2)
Subject Accessibility	88.4±3.6 (wt=2)	63.9±5.0 (wt=2)	74.2±5.1 (wt=2)
Restricted Memory	87.5±3.0 (limit=25)	61.5±5.5 (limit=25)	72.8±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}

8.3.2 Combining cognitive biases

Table 16 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 and for a fairly large number of biases and parameter settings would have taken approximately 10 days on a Sparc20 for each data set. Rather than limit the number of biases and parameter settings considered, however, we chose to limit the number of partitions used in the inner cross validation that selects the biases and parameter settings for a single test partition (Step 1(b)i of figure 1): All results in table 16 were obtained with 10 outer partitions and two of the ten inner partitions. 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 perfor-

Table 16: 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
p-o-s	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

mance over the recency bias. Table 17 shows why: Recency is often the only bias selected for the lexical tagging tasks. This is the case in seven out of 10 partitions for both semantic tagging tasks and indicates that the features associated with neighboring words are critical for the lexical tagging tasks. The restricted memory bias is also selected for part-of-speech tagging, but not for the semantic tagging tasks. The effect of this bias is to discard all 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 — for both tasks, the recency bias provides additional weight primarily for features associated with a three-word window centered on the unknown word. (A maximum recency weight of 11 increases the weights representing the morphology of the unknown word and the five features associated with each of the preceding and following tokens.) 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.

Table 17: Summary of greedy bias selection results. 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 = {3, 6, 8, 14}
Restricted Memory: limit = {3, 6, 8, 11, 14, 21, 25}

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 one major difference between the bias selection algorithms, however: the exhaustive approach tends to include more biases than the greedy approach. For example, the restricted memory bias is selected for general semantic class tagging for eight out of 10 folds when it was never selected by the greedy algorithm. The syntactic bias was selected for part-of-speech tagging for eight out of 10 folds even though it was rarely (one time) selected by the greedy algorithm. One or both of the syntactic and semantic biases (wt = 2) were selected in six out of 10 folds for general semantic class tagging; they were selected (wt = 2 or 4) in seven out of 10 folds for specific semantic tagging.

Overall, our results appear to indicate that, given the IE system'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 selects biases as well as the much more expensive exhaustive bias selection algorithm.

9 Discussion and related work

Natural Language Learning. We have tackled four natural language learning tasks in this work: relative pronoun resolution and predicting the part-of-speech, and general and specific semantic features for unknown words. There has been prior, related work on each of these tasks. First, there is a growing body of work on pronoun resolution. Techniques for pronoun resolution generally operate through the application of a set of linguistically-based, hand-crafted filters that vary depending on the type of pronoun (e.g. Lappin and Leass (1994)). Others rely on focus-based approaches that incorporate discourse information when resolving pronouns (e.g. Azzam et al. (1998)). The goal of our approach, on the other hand, is to avoid manual knowledge base construction altogether. In practice, hand-crafted heuristics need to be tuned as the underlying linguistic knowledge sources and the domain of the natural language application change. By learning heuristics for pronoun resolution automatically, system developers can avoid the tedium and difficulties associated with knowledge-based approaches to this important natural language understanding task.

Corpus-based approaches to pronoun resolution, however, are beginning to appear. Ge et al. (1998) , for example, present a supervised probabilistic algorithm that assumes a full parse of the input text. Dagan and Itai (1991) present a hybrid full-parse/unsupervised learning approach that focuses on resolving 'it'. Despite a large corpus (150 million words), their approach suffers from sparse data problems, but works well when enough relevant data is available. To date, however, we know of no similar work for relative pronoun resolution.

Closely related to pronoun resolution is the more general problem of noun phrase coreference. Like pronoun resolution, the vast majority of noun phrase coreference algorithms combine syntactic and, less often, semantic cues via a set of hand-crafted heuristics and filters. All but one system in the MUC-6 coreference performance evaluation (MUC-6 1995), for example, handled coreference resolution in this manner. Recently, however, learning approaches to noun phrase coreference resolution have been proposed (e.g. Aone and Bennett (1995), McCarthy and Lehnert (1995) and Kehler (1997). By recasting the coreference problem as a classification task, any of a number of standard inductive learning algorithms can be employed: Given two phrases and the contexts in which they occur, the learning algorithm must classify the phrases with respect to whether or not they refer to the same object. Training sets consist of one instance for every possible pairing of referents in the training texts: some of these are 'positive' examples in that they correspond to phrases that *are* coreferent; others are 'negative' examples in that they correspond to phrases that are *not* referring to the same object. Existing machine learning approaches to coreference resolution typically use decision tree induction algorithms (Quinlan 1993) as the inductive learning component (Aone and Bennett 1995, McCarthy and Lehnert 1995). A separate mechanism then decides among all phrases selected by the learning algorithm as likely referents. Without exception, corpus-based methods for noun phrase coreference resolution have been designed primarily for use in information extraction systems: McCarthy's RESOLVE system (1995) and Aone and Bennett's MLR system (1995) both operate on texts that describe business joint ventures and focus on coreference involving organizations (e.g. companies, governments).

In contrast, our case-based approach to learning heuristics for relative pronoun resolution does not require testing all pairings of the relative pronoun with potential noun phrase antecedents. Instead, it selects the referent in one step by examining the preceding context of the relative pronoun. Nonetheless, the general coreference problem is a much broader task than relative pronoun resolution; it is unclear whether the approach for relative pronoun resolution presented here could be successfully extended to this more general task. In particular, our assumption that a fairly local representation of context is sufficient for anaphora resolution would quickly be violated for general coreference resolution. In recent work, we have instead explored the use of unsupervised techniques that view noun phrase coreference as clustering (Cardie and Wagstaff 1999).

Similarly, our work on part-of-speech assignment is related, but not directly comparable, to the growing body of work on part-of-speech tagging. In the work presented here, we focus on learning lexical information for unknown words only, which is generally considered a more difficult task. Note, however, that case-based learning algorithms have been evaluated on the standard part-of-speech tagging task, and have been shown to perform as well as the best statistical and rule-based methods (Cardie 1996, Daelemans et al. 1996).

In addition, there has been recent work on semantic feature acquisition (Riloff 1997, Roark and Charniak 1998). The work focused, however, on learning a small, static semantic lexicon rather than dynamically tagging all words in an incoming text. Viewed slightly differently, our work on semantic class tagging can be seen as a type of word-sense disambiguation — a task that has been addressed empirically by a number of researchers (e.g. Yarowsky (1992) and Bruce and Wiebe (1994)). The word-sense disambiguation task, however, normally focuses on the disambiguation of all occurrences of a small set of polysemous words in a large corpus. Our work differs in two primary ways. First, we focus on domain-specific word-sense disambiguation, i.e. word-sense disambiguation with respect to a semantic class hierarchy defined with respect to a particular end-to-end NLP task. Second, instead of evaluation on a small set of words, we evaluate the system's ability to assign semantic features to *all* words in an input text. Again, however, case-based approaches to traditional word-sense disambiguation have been very successful (Ng and Lee 1996).

Thus far, we have tested the cognitive bias approach to feature set selection on just three lexical disambiguation tasks and one structural disambiguation task in conjunction with a single NLP system. To determine the generality of the approach for natural language learning, future work should include applying the methods presented here to additional natural language learning problems embedded in a variety of NLP systems.

In this work, we exploit existing results from psycholinguistics to inform the case-based learning algorithm. In the long run, however, we would hope that our computational models could likewise inform the design of psycholinguistic experiments. Currently, our approach only coarsely models the effect of cognitive processing preferences and cognitive bias interactions on language processing tasks. As the computational models become increasingly sophisticated, however, we might use them to aid in the selection of sentences for human studies or to present language processing hypotheses for further study.

Feature Selection and Feature Weighting for CBL. Because feature selection and feature weighting are critical for the success of case-based learning algorithms, much research has addressed these problems. However, we know of no previous work that makes use of cognitive biases to guide either process. 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. Local feature-weighting schemes allow feature weights to vary from instance to instance, or feature value to feature value. These methods work best when features vary in importance across the instance space (Aha and Goldstone 1992). Many such local weighting methods for case-based learning have been proposed. The value difference metric of Stanfill & Waltz (1986), for example, was an early machine learning algorithm that assigned a different weight to each value of a feature. In other work, Aha & 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 is one example (Howe and 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) uses per-category feature importance (PCF) to assign high weight to features that are highly correlated with the class. Aha's IB4 classifier also calculates a different weight vector for each class (Aha 1992). 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 for some of the same natural language learning tasks presented here (Cardie and Howe 1997). Hastie & 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 in a (continuous) function-learning paradigm rather than a classification paradigm. Note that 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 to generate a weight for each feature in the test case. 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 et al. 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 Cobweb system (1987) 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 multivariate 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. In earlier work (Cardie 1992), we implemented a partially automated version of the cognitive bias approach for Cobweb and obtained similar results to those reported here.

Global feature weighting methods that compute weights based on information gain have proved particularly helpful across a variety of natural language processing tasks. Daelemans et al. (1999), for example, obtain excellent results for a broad range of language-learning problems by using the information gain associated with each feature as real-valued feature weights. In related work, we have used decision trees to discard irrelevant features from an instance representation (Cardie 1993b). The idea is simply to present the training cases as input to a decision tree induction algorithm. Any features that appear in the pruned decision tree then become part of the case representation used by the case-based learning algorithm; all other features are discarded. We combined these two algorithms to create the IG-CBL feature-weighting algorithm (Cardie and Howe 1997) that was applied to the four natural language learning tasks in earlier sections.

A drawback of all of the information gain techniques is that the methods work poorly for data sets where information gain does not discriminate among features. They also encounter problems when complex feature interactions exist and can not be adequately reflected in the information gain computations. This situation is more likely to arise as the case representation expands to encode state information for increasingly sophisticated NLP systems. IG-CBL worked particularly poorly for our relative pronoun data set. One possible reason for this is that IG-CBL requires the case representation to be normalized, i.e. every case must include a value for every feature that appeared in the training set. As a result, cases have a large number of features, most of which correspond to constituents that did not appear in the sentence from which the case was derived. One avenue for future work is to combine information-based feature selection methods with the cogntive bias-based approaches presented here.

An advantage of the general feature-weighting techniques described above is that they are entirely task-independent — they can be applied without modification to any learning problem that is encoded in the case-based learning framework. Our approach is useful only for tasks where cognitive biases make sense, as in the language processing tasks described here. In addition, porting our approach for use with a new NLP system will require the re-implementation of all cognitive biases to reflect system-specific characteristics.

Furthermore, the success of the cognitive bias feature weighting algorithm depends, in part, on the ability of the bias selection algorithm to choose the bias combination that is appropriate for the learning task. Unfortunately, the computational complexity of this task forces us to trade a potential loss in accuracy for an increase in speed — by evaluating bias combinations using a small number of inner cross validation partitions rather than nested leave-one-out cross validation. We may have seen larger effects for some of the tested biases if the full cross validation scheme had been executed. As noted elsewhere, the size of the data sets used in our studies may have also limited the effects of some of the biases and future work must include evaluation of the cognitive bias approach on larger data sets.

Bias Selection. 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 a survey article 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. Both search and cross validation have been used successfully for a variety of problems in machine learning. Greedy search algorithms, for example, have been applied to the problem of feature set selection (e.g. Caruana and Freitag (1994), John et al. (1994), Skalak (1994), Xu et al. (1989)). More specifically, forward selection of features begins by selecting the best individual feature for the instance representation from the set of available features; in each subsequent iteration, one more feature is selected until performance degrades. Backward selection begins with an instance representation that contains all available features; in each subsequent iteration, one feature is selected for removal. Again, the process continues until performance degrades. An advantage of these feature selection algorithms is that they are finer-grained than the cognitive bias approach proposed here — they add or remove one feature at a time. The methods, however, are slow; they may not be feasible for tasks with large feature sets. An advantage of our approach is that search occurs not in the feature space, but in the much smaller bias space.

Nevertheless, a major problem with the cognitive bias selection algorithms, especially exhaustive bias selection, is speed: as the data sets increase in size and the number of biases increases, the approach becomes less feasible. As a result, one 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. Incorporating standard approaches for shortcutting the cross-validation process (Moore and Lee 1994) may also provide substantial decreases in training time.

Cross validation has been used extensively in statistics and machine learning as a standard technique for model selection and parameter setting. Schaffer (1993), for example, uses 10-fold cross validation to select the best of a sequence of increasingly pruned decision trees. Cross validation has also been employed successfully for feature set selection (e.g. Moore and Lee (1994)). Our work differs from this previous work in that we use search and cross validation for bias selection (rather than feature set selection). The selected bias or biases are then responsible for directing feature set selection and feature weighting.

10 Summary

This paper presents a new technique for feature set selection for case-based learning of linguistic knowledge. The approach employs cognitive processing preferences as a domain-independent mechanism to guide feature set selection. We have applied the method to four natural language learning tasks 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 with appropriate parameter settings for each natural language learning task: the case representation that reflects the selected cognitive bias or bias combination significantly outperforms the baseline case representation as well as the default heuristics for each data set.

Moreover, our experiments on relative pronoun disambiguation indicate that the case-based learning algorithm improves as additional 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.

In summary, we believe that the cognitive bias approach to feature set selection offers new possibilities for case-based learning of natural language. It simplifies the process of designing an appropriate instance representation for individual natural language learning tasks because system developers can safely include in the baseline instance representation features for all available knowledge sources. In the long run, it may obviate the need for separate instance representations for each linguistic knowledge acquisition task. More importantly, the cognitive bias approach to case-based learning of natural language offers a mechanism for explicitly combining the frequency information available from corpus-based techniques with linguistic bias information employed in traditional linguistic and knowledge-based approaches to natural language processing. By creating computational models of cognitive biases, we may also be able to offer linguists and psycholinguists insights into the relative importance of existing cognitive and linguistic biases for individual natural language processing phenomena.

11 Acknowledgments

This work was supported in part by NSF Grant IRI–9624639.

References

D. W. Aha. 1992. Tolerating Noisy, Irrelevant, and Novel Attributes in Instance-Based Learning Algorithms. *International Journal of Man-Machine Studies*, 36:267–287.

D. W. Aha and R. L. Goldstone. 1992. Concept learning and flexible weighting. In *Proceedings of the Fourteenth Annual Conference of the Cognitive Science Society*, pages 534–539, Indiana Univeristy, Bloomington, IN. Lawrence Erlbaum Associates.

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

Chinatsu Aone and William Bennett. 1995. Evaluating Automated and Manual Acquisition of Anaphora Resolution Strategies. In *Proceedings of the 33rd Annual Meeting of the ACL*, pages 122–129. Association for Computational Linguistics.

S. Argamon, I. Dagan, and Y. Krymolowski. 1998. A Memory-Based Approach to Learning Shallow Natural Language Patterns. In *Proceedings of the 36th Annual Meeting of the ACL and COLING-98*, pages 67–73. Association for Computational Linguistics.

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

S. Azzam, K. Humphreys, and R. Gaizauskas. 1998. Evaluating a Focus-Based Approach to Anaphora Resolution. In *Proceedings of the 36th Annual Meeting of the ACL and COLING-98*, pages 74–78. Association for Computational Linguistics.

R. Bruce and J. Wiebe. 1994. Word-Sense Disambiguation Using Decomposable Models. In *Proceedings of the 32th Annual Meeting of the ACL*, pages 139–146.

T. Cain, M. Pazzani, and G. Silverstein. 1991. Using domain knowledge to influence similarity judgement. In *Proceedings of the Case-Based Reasoning Workshop*, pages 191–199. Morgan Kaufmann.

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

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

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

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

C. Cardie. 1996. Embedded machine learning systems for natural language processing: A general framework. In Stefan Wermter, Ellen Riloff, and Gabriele Scheler, editors, *Symbolic, connectionist, and statistical approaches to learning for natural language processing*, Lecture Notes in Artificial Intelligence Series, pages 315–328. Springer.

C. Cardie. 1997. Empirical Methods in Information Extraction. *AI Magazine*, 18(4):65–79.

C. Cardie. 1999. A Cognitive Bias Approach to Feature Set Selection. *Machine Learning*. To appear.

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

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

C. Cardie and K. Wagstaff. 1999. Noun Phrase Coreference as Clustering. In *Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-99)*, pages 82–89. ACL SIGDAT.

M. Carreiras, M. A. Gernsbacher, and V. Villa. 1995. The Advantage of First Mention in Spanish. *Psychonomic Bulletin and Review*, 2:124–129.

R. Caruana and D. Freitag. 1994. Greedy Attribute Selection. In *Proceedings of the Eleventh International Conference on Machine Learning*, pages 28–36, Rutgers University, New Brunswick, NJ. Morgan Kaufmann.

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

C. Clifton, L. Frazier, and C. Connine. 1984. Lexical expectations in sentence comprehension. *Journal of Verbal Learning and Verbal Behavior*, 23:696–708.

S. Crain and M. Steedman. 1985. On not being led up the garden path: The use of context by the psychological syntax processor. In D. Dowty, L. Karttunen, and A. Zwicky, editors, *Natural Language Parsing: Psychological, computational, and theoretical perspectives*, pages 320–358. Cambridge University Press.

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

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

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

W. Daelemans, A. van den Bosch, and T. Weijters. 1997. IGTree: Using Trees for Compression and Classification in Lazy Learning Algorithms. *Artificial Intelligence Review*, 11:407–423.

W. Daelemans, A. van den Bosch, and J. Zavrel. 1999. Forgetting Exceptions is Harmful in Language Learning. *Machine Learning*, 34(1–3):11–43.

W. Daelemans, J. Zavrel, Berck P., and Gillis S. 1996. MBT: A Memory-Based Part of Speech Tagger Generator. In *Proceedings of the Fourth Workshop on Very Large Corpora*, pages 14–27. ACL SIGDAT.

I. Dagan and A. Itai. 1991. A Statistical Filter for Resolving Pronoun References. In Y. A. Feldman and A. Bruckstein, editors, *Artificial Intelligence and Computer Vision*, pages 125–135. Elsevier Science Publishers, North Holland.

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

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

T. Dietterich. 1998. Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms. *Neural Computation*, 10(7):1895–1924.

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

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

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

J. Fodor. 1989. Empty Categories in Sentence Processing. *Language and Cognitive Processes*, 4(3-4):155–209.

L. Frazier. 1979. On comprehending sentences: Syntactic parsing strategies. Indiana University Linguistics Club, Bloomington, IN.

L. Frazier. 1991. Parsing modifiers: Special purpose routines in the human sentence processing mechanism? In D. A. Balota, G.B. Flores d'Arcais, and K. Rayner, editors, *Comprehension processes in reading*, pages 303–330. Lawrence Erlbaum Associates, Hillsdale, NJ.

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

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

N. Ge, J. Hale, and E. Charniak. 1998. A Statistical Approach to Anaphora Resolution. In *Proceedings of the Sixth Workshop on Very Large Corpora*, pages 161–170, Montreal, Canada. ACL SIGDAT.

M. A. Gernsbacher and D. J. Hargreaves. 1988. Accessing Sentence Participants: The Advantage of First Mention. *Journal of Memory and Language*, 27:199–717.

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

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

E. Gibson. 1991. *A computational theory of human linguistic processing: Memory limitations and processing breakdown*. Ph.D. thesis, Carnegie Mellon University.

E. Gibson and N. Pearlmutter. 1994. A Corpus-Based Analysis of Psycholinquistic Constraints on Prepositional-Phrase Attachment. In C. Clifton Jr., L. Frazier, and K. Rayner, editors, *Perspectives on Sentence Processing*, pages 181–198, Hillsdale, NJ. Lawrence Erlbaum Associates.

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

D. Gordon and M. desJardins. 1995. Evaluation and Selection of Biases in Machine Learning. *Machine Learning*, 20(1–2):5–22.

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

N. Howe and C. Cardie. 1997. Examining Locally Varying Weights for Nearest Neighbor Algorithms. In *Proceedings of the Second International Conference on Case-Based Reasoning*, pages 455–466, Providence, RI. Springer.

G. John, R. Kohavi, and K. Pfleger. 1994. Irrelevant Features and the Subset Selection Problem. In *Proceed-*

ings of the Eleventh International Conference on Machine Learning, pages 121–129, Rutgers University, New Brunswick, NJ. Morgan Kaufmann.

A. Kehler. 1997. Probabilistic Coreference in Information Extraction. In *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*, pages 163–173. Association for Computational Linguistics.

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

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

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

S. Lappin and H. Leass. 1994. An Algorithm for Pronominal Anaphora Resolution. *Computational Linguistics*, 20(4):535–562.

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

W. Lehnert and B. Sundheim. 1991. A performance evaluation of text analysis technologies. *AI Magazine*, 12(3):81–94.

W. Lehnert, C. Cardie, D. Fisher, J. McCarthy, E. Riloff, and S. Soderland 1992. University of Massachusetts: Description of the CIRCUS System as Used in MUC-4. In *Proceedings of the Fourth Message Understanding Conference (MUC-4)*, pages 282–288, San Mateo, CA. Morgan Kaufmann.

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

W. Lehnert, J. McCarthy, S. Soderland, E. Riloff, C. Cardie, J. Peterson, F. Feng, C. Dolan, and S. Goldman. 1993. University of Massachusetts/Hughes: Description of the CIRCUS System as Used in MUC-5. In *Proceedings of the Fifth Message Understanding Conference (MUC-5)*, pages 277–291, San Mateo, CA. Morgan Kaufmann.

M. C. MacDonald, N. J. Pearlmutter, and M. S. Seidenberg. 1994. Syntactic Ambiguity Resolution as Lexical Ambiguity Resolution. In C. Clifton Jr., L. Frazier, and K. Rayner, editors, *Perspectives on Sentence Processing*, pages 123–153. Lawrence Erlbaum Associates, Hillsdale, NJ.

J. McCarthy and W. Lehnert. 1995. Using Decision Trees for Coreference Resolution. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence*, pages 1050–1055.

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

A. Moore and M. Lee. 1994. Efficient Algorithms for Minimizing Cross Validation Error. In *Proceedings of the Eleventh International Conference on Machine Learning*, pages 190–198, Rutgers University, New Brunswick, NJ. Morgan Kaufmann.

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

MUC-4. 1992. *Proceedings of the Fourth Message Understanding Conference (MUC-4)*. Morgan Kaufmann, San Mateo, CA.

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

MUC-6. 1995. *Proceedings of the Sixth Message Understanding Conference (MUC-6)*. Morgan Kaufmann, San Francisco, CA.

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

H. Ng and H. Lee. 1996. Integrating Multiple Knowledge Sources to Disambiguate Word Sense: An Exemplar-Based Approach. In *Proceedings of the 34th Annual Meeting of the ACL*, pages 40–47. Association for Computational Linguistics.

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

J. R. Quinlan. 1983. Learning Efficient Classification Procedures and Their Application to Chess End

Games. In R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, editors, *Machine Learning: An Artificial Intelligence Approach*. Morgan Kaufmann, San Mateo, CA.

J. R. Quinlan. 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA.

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

E. Riloff. 1997. A Corpus-Based Approach for Building Semantic Lexicons. In *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*, pages 117–124. Association for Computational Linguistics.

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

B. Roark and E. Charniak. 1998. Noun-phrase co-occurrence statistics for semi-automatic semantic lexicon construction. In *Proceedings of the 36th Annual Meeting of the ACL and COLING-98*, pages 1110–1116. Association for Computational Linguistics.

C. Schaffer. 1993. Overfitting Avoidance as Bias. *Machine Learning*, 10(2):153–178.

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

G. Simpson and M. Krueger. 1991. Selective access of homograph meanings in sentence context. *Journal of Memory and Language*, 30:627–643.

D. Skalak. 1992. Representing cases as knowledge sources that apply local similarity metrics. In *Proceedings of the Fourteenth Annual Conference of the Cognitive Science Society*, pages 325–330, Indiana Univeristy, Bloomington, IN. Lawrence Erlbaum Associates.

D. Skalak. 1994. Prototype and Feature Selection by Sampling and Random Mutation Hill Climbing Algorithms. In *Proceedings of the Eleventh International Conference on Machine Learning*, pages 293–301, Rutgers University, New Brunswick, NJ. Morgan Kaufmann.

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

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

D. Swinney, M. Ford, J. Bresnan, and U. Frauenfelder. 1988. Coreference assignment during sentence processing. In M. Macken, editor, *Language structure and processing*. CSLI, Stanford, Palo Alto, CA.

M. Tanenhaus, J. Boland, S. Garnsey, and G. Carlson. 1989a. Lexical Structure in Parsing Long-Distance Dependencies. *Journal of Psycholinguistic Research*, 18(1):37–50.

M. Tanenhaus, G. Carlson, and J. C. Trueswell. 1989b. The Role of Thematic Structures in Interpretation and Parsing. *Language and Cognitive Processes*, 4:211–234.

A. van den Bosch and W. Daelemans. 1993. Data-oriented methods for grapheme-to-phoneme conversion. In *Proceedings of the Sixth Conference of the EACL*, pages 45–53, Utrecht. Also available as ITK Research Report 42.

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

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

D. Yarowsky. 1992. Word-Sense Disambiguation Using Statistical Models of Roget's Categories Trained on Large Corpora. In *Proceedings, COLING-92*, pages 454–460. ICCL.

J. Zavrel, W. Daelemans, and J. Veenstra. 1997. Resolving pp attachment ambiguities with memory-based learning. In *Proceedings of the Workshop on Computational Language Learning (CoNLL '97)*, pages 136–144. Association for Computational Linguistics.