Learning Methods to Combine Linguistic Indicators: Improving Aspectual Classification and Revealing Linguistic Insights

Eric V. Siegel*
Columbia University

Kathleen R. McKeown†
Columbia University

Aspectual classification maps verbs to a small set of primitive categories in order to reason about time. This classification is necessary for interpreting temporal modifiers and assessing temporal relationships, and is therefore a required component for many natural language applications.

A verb's aspectual category can be predicted by co-occurrence frequencies between the verb and certain linguistic modifiers. These frequency measures, called linguistic indicators, are chosen by linguistic insights. However, linguistic indicators used in isolation are predictively incomplete, and are therefore insufficient when used individually.

In this article, we compare three supervised machine learning methods for combining multiple linguistic indicators for aspectual classification: decision trees, genetic programming, and logistic regression. A set of 14 indicators are combined for classification according to two aspectual distinctions. This approach improves the classification performance for both distinctions, as evaluated over unrestricted sets of verbs occurring across two corpora. This demonstrates the effectiveness of the linguistic indicators and provides a much-needed full-scale method for automatic aspectual classification. Moreover, the models resulting from learning reveal several linguistic insights that are relevant to aspectual classification. We also compare supervised learning methods with an unsupervised method for this task.

1. Introduction

Aspectual classification maps clauses (e.g., simple sentences) to a small set of categories in order to reason about time. For example, **events**, such as, *You called your father*, are distinguished from **states**, such as, *You resemble your father*. The ability to distinguish stative clauses from event clauses is a fundamental component of natural language understanding. These two high-level categories correspond to fundamental distinctions in many domains, including the distinctions between diagnosis and procedure in the medical domain, and between analysis and activity in the financial domain.

Stativity is the first high-level distinction made when defining the **aspectual class** of a clause. Events are further distinguished according to **completedness** (sometimes called **telicity**), which determines whether an event reaches a culmination or completion point at which a new state is introduced. For example, *I made a fire* is **culminated**, since a new state is introduced—something is made, whereas *I gazed at the sunset* is **nonculminated**.

* Computer Science Dept., 1214 Amsterdam Ave., New York, NY 10027. E-mail: evs@cs.columbia.edu
† Computer Science Dept., 1214 Amsterdam Ave., New York, NY 10027. E-mail: kathy@cs.columbia.edu

Aspectual classification is necessary for interpreting temporal modifiers and assessing temporal entailments (Moens and Steedman 1988; Dorr 1992; Klavans 1994) and is therefore a required component for applications that perform certain natural language interpretation, generation, summarization, information retrieval, and machine translation tasks. Each of these applications requires the ability to reason about time.

A verb's aspectual category can be predicted by co-occurrence frequencies between the verb and linguistic phenomena such as the progressive tense and certain temporal modifiers (Klavans and Chodorow 1992). These frequency measures are called **linguistic indicators**. The choice of indicators is guided by linguistic insights that describe how the aspectual category of a clause is constrained by the presence of these modifiers. For example, an event can be placed in the progressive, as in, *She was jogging*, but many stative clauses cannot, e.g., **She was resembling her father* (Dowty 1979). One advantage of linguistic indicators is that they can be measured automatically.

However, individual linguistic indicators are predictively incomplete, and are therefore insufficient when used in isolation. As demonstrated empirically in this article, individual linguistic indicators suffer from limited classification performance due to several linguistic and pragmatic factors. For example, some indicators were not motivated by specific linguistic insights. However, linguistic indicators have the potential to interact and supplement one another, so it would be beneficial to combine them systematically.

In this article, we compare three supervised machine learning methods for combining multiple linguistic indicators for aspectual classification: decision trees, genetic programming, and logistic regression. A set of 14 indicators are combined, first for classification according to stativity, and then for classification according to completedness. This approach realizes the potential of linguistic indicators, improving classification performance over a baseline method for both tasks with minimal overfitting, as evaluated over an unrestricted set of verbs occurring in two corpora. This serves to demonstrate the effectiveness of these linguistic indicators and thus provides a much-needed full-scale, expandable method for automatic aspectual classification.

The results of learning are linguistically viable in two respects. First, learning automatically produces models that are specialized for different aspectual distinctions; the same set of 14 indicators are combined in different ways according to which classification problem is targeted. Second, inspecting the models resulting from learning revealed linguistic insights that are relevant to aspectual classification.

We also evaluate an unsupervised method for this task. This method uses co-occurrence statistics to group verbs according to meaning. Although this method groups verbs generically and is not designed to distinguish according to aspectual class in particular, we show that the results do distinguish verbs according to stativity.

The next two sections of this article describe aspectual classification and linguistic indicators. Section 4 describes the three supervised learning methods employed to combine linguistic indicators for aspectual classification. Section 5 gives results in terms of classification performance and resulting linguistic insights, comparing these results, across classification tasks, to baseline methods. Section 6 describes experiments with an unsupervised approach. Finally, Sections 7, 8, and 9 survey related work, describe future work, and present conclusions.

2. Aspect in Natural Language

Because, in general, the sequential order of clauses is not enough to determine the underlying chronological order, aspectual classification is required for interpreting

Table 1
Aspectual classes. This table is adapted from Moens and Steedman (1988, p. 17).

	EVENTS		STATES
	punctual	extended	
Culminated	CULMINATION	CULMINATED PROCESS	
	recognize	build a house	understand
Nonculminated	POINT	PROCESS	
	hiccup	run, swim, walk	

even the simplest narratives in natural language. For example, consider:

(1) Sue mentioned Miami (event). Jim cringed (event).

In this case, the first sentence describes an event that takes place before the event described by the second sentence. However, in

(2) Sue mentioned Miami (event). Jim already knew (state).

the second sentence describes a state, which begins before the event described by the first sentence.

Aspectual classification is also a necessary prerequisite for interpreting certain adverbial adjuncts, as well as identifying temporal constraints between sentences in a discourse (Moens and Steedman 1988; Dorr 1992; Klavans 1994). In addition, it is crucial for lexical choice and tense selection in machine translation (Moens and Steedman 1988; Klavans and Chodorow 1992; Klavans 1994; Dorr 1992).

Table 1 summarizes the three aspectual distinctions, which compose five aspectual categories. In addition to the two distinctions described in the previous section, **atomicity** distinguishes **punctual** events (e.g., *She noticed the picture on the wall*) from **extended** events, which have a time duration (e.g., *She ran to the store*). Therefore, four classes of events are derived: **culmination**, **culminated process**, **process**, and **point**.

These aspectual distinctions are motivated by a series of syntactic and entailment constraints described in the first three subsections below. Further cognitive and philosophical rationales behind these semantic distinctions are surveyed by Siegel (1998b). First we describe aspectual constraints that linguistically motivate the design of several of the linguistic indicators. Next we describe an array of semantic entailments and temporal constraints that can be put to use by an understanding system once input clauses have been aspectually classified. Then we describe how aspect influences the interpretation of temporal connectives and modifiers. Aspectual transformations are described, and we introduce the concept of a clause's **fundamental** aspectual category. Finally, we describe several natural language applications that require an aspectual classification component.

2.1 Aspectual Markers and Constraints

Certain features of a clause, such as the presence of adjuncts and tense, are constrained by and contribute to the aspectual class of the clause (Vendler 1967; Dowty 1979; Pustejovsky 1991; Passonneau 1988; Klavans 1994; Resnik 1996; Olsen and Resnik 1997). Table 2 illustrates an array of linguistic constraints, as more comprehensively

Table 2
Several aspectual markers and associated constraints on aspectual class.

If a clause can occur:	**then it must be:**
with a temporal adverb (e.g., *then*)	Event
in progressive	Extended Event
as a complement of *force/persuade*	Event
after "*What happened was. . .*"	Event
with a duration *in*-PP (e.g., *in an hour*)	Culminated Event
in the perfect tense	Culminated Event or State

summarized by Klavans (1994) and Siegel (1998b). Each entry in this table describes an aspectual **marker** and the constraints on the aspectual category of any clause that appears with that marker. For example, a clause must be extended to appear in the progressive tense, e.g.,

(3) He was prospering in India (extended event),

which contrasts with,

(4) *You were noticing that I can hardly be blamed . . . (atomic event).[1]

As a second example, since the perfect tense requires that the clause it occurs in must entail a consequent state, an event must be culminated to appear in the perfect tense. For example,

(5) Thrasymachus has made an attempt to get the argument into his own hands (culminated event),

contrasts with,

(6) *He has cowered down in a paralysis of fear (nonculminated event).

2.2 Aspectual Entailments

Table 3 lists several aspectual entailments. A more comprehensive list can be found in Klavans (1994) or Siegel (1998b). Each entry in this table describes a linguistic phenomenon, a resulting entailment, and the constraints on aspectual class that apply if the resulting entailment holds. For example, the simple present reading of an event, e.g., *He jogs*, denotes the **habitual** reading, i.e., *every day*, whereas the simple present reading of a state, e.g., *He appears healthy*, entails *at the moment*.

These entailments serve two purposes: They further validate the three aspectual distinctions, and they illustrate an array of inferences that can be made by an understanding system. However, these inferences can only be made after identifying the aspectual category of input clauses.

1 These example sentences are modifications of samples from the corpus of novels described below.

Table 3
Several aspectual entailments.

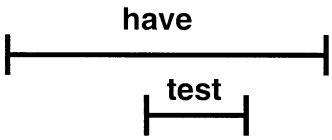
If a clause occurring:	necessarily entails:	then it must be:
in past progressive tense	past tense reading	Nonculminated Event
as argument of *stopped*	past tense reading	Nonculminated Event or State
in simple present tense	the habitual reading	Event

2.3 Interpreting Temporal Connectives and Modifiers

Several researchers have developed models that incorporate aspectual class to assess temporal constraints between connected clauses (Hwang and Schubert 1991; Schubert and Hwang 1990; Dorr 1992; Passonneau 1988; Moens and Steedman 1988; Hitzeman, Moens, and Grover 1994). For example, stativity must be identified to detect temporal constraints between clauses connected with *when*. For example, in interpreting,

(7) She **had** good strength when objectively **tested**.[2]

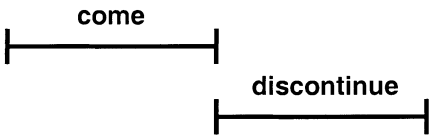
the *have* state began before or at the beginning of the *test* event, and ended after or at the end of the *test* event:



However, in interpreting,

(8) Phototherapy was **discontinued** when the bilirubin **came** down to 13.

the *discontinue* event began at the end of the *come* event:



Such models also predict temporal relations between clauses combined with other connectives such as *before*, *after*, and *until*.

Certain temporal modifiers are disambiguated with aspectual class. For example, *for an hour* can denote the duration of a nonculminated event, as in, *I gazed at the sunset* **for an hour**. In this case, *an hour* is the duration of the *gazing* event. However, when applied to a culminated event, it denotes the duration of the resulting state, as in, *I left the room* **for an hour**. In this case, *an hour* is not the duration of the *leaving* event, but, rather, the duration of what resulted from *leaving*, i.e., being gone.

2 These examples of *when* come from the corpus of medical discharge summaries described below.

2.4 Aspectual Transformations and Coercion

Several aspectual markers such as those shown in Table 2 transform the aspectual class of the clause they modify. For example, a duration *for*-PP, e.g., *for 10 minutes*, denotes the time duration of a (nonculminated) process, resulting in a culminated process, e.g.,

(9) I stared at it (process).

(10) I stared at it for 10 minutes (culminated process).

Some **aspectual auxiliaries** also perform an aspectual transformation of the clause they modify, e.g.,

(11) I finished staring at it (culminated process).

Aspectual coercion, a second type of aspectual transformation, can take place when a clause is modified by an aspectual marker that violates an aspectual constraint (Moens and Steedman 1988; Pustejovsky 1991). In this case, an alternative interpretation of the clause is inferred which satisfies the aspectual constraint. For example, the *progressive* marker is constrained to appear with an extended event. Therefore, if it appears with an atomic event, e.g.,

(12) He hiccupped (point),

the event is transformed to an extended event, e.g.,

(13) He was hiccupping (process).

in this case with the **iterated** reading of the clause (Moens and Steedman 1988).

2.5 The First Problem: Fundamental Aspect

We define **fundamental aspectual class** as the aspectual class of a clause *before* any aspectual transformations or coercions. That is, the fundamental aspectual category is the category the clause would have if it were stripped of any and all aspectual markers that induce an aspectual transformation, as well as all components of the clause's pragmatic context that induce a transformation. Fundamental aspectual class is therefore a function of the main verb and a select group of complements, as illustrated in the previous two subsections. It is the task of detecting fundamental aspect that we address in this article. As established by some previous work in linguistics, adjuncts are to be handled separately from other clausal constituents in assessing aspectual class (Pustejovsky 1995).

An understanding system can recognize the aspectual transformations that have affected a clause only after establishing the clause's fundamental aspectual category. Linguistic models motivate the division between a module that first detects fundamental aspect and a second that detects aspectual transformations (Hwang and Schubert 1991; Schubert and Hwang 1990; Dorr 1992; Passonneau 1988; Moens and Steedman 1988; Hitzeman, Moens, and Grover 1994). In principle, it is possible for this second module to detect aspectual transformations that apply to any input clause, independent of the manner in which the core constituents interact to produce its fundamental aspectual class.

2.6 Applications of Aspectual Classification

Aspectual classification is a required component of applications that perform natural language interpretation, natural language generation, summarization, information retrieval, and machine translation tasks (Moens and Steedman 1988; Klavans and Chodorow 1992; Klavans 1994; Dorr 1992; Wiebe et al. 1997). These applications require the ability to reason about time, i.e., **temporal reasoning**.

Assessing temporal relationships is a prerequisite for inferring sequences of medical procedures in medical domains. Many applications that process medical reports require aspectual classification because a patient's medical progress and history are established as a series of states and events that are temporally related. One task is to automatically complete a database entry for the patient by processing a medical discharge summary detailing a patient's visit to the hospital. For example, consider the temporal relationship between the clauses connected with *when* in,

(14) The small bowel **became** completely free when dissection was **continued**.[3]

In this case, the *become* culmination takes place at the onset of the *continue* process. However, in

(15) The small bowel **became** completely free when dissection was **performed**.

the *become* culmination takes place at the completion of the *perform* culminated process.

Aspect is also crucial for tense selection in machine translation between certain pairs of languages because some languages have explicit perfective markers and others do not. The perfective marker is used in many languages, such as Bulgarian and Russian, to indicate completedness. Therefore, a system translating from a language without explicit perfective markers, such as English, to one with explicit perfective markers must first detect the aspectual category of an input phrase in order to determine the form of the output (Stys 1991; Dorr 1992). Aspect is also required for lexical selection in machine translation since, for example, some languages, e.g., German and French, have different words for the two uses of *for* discussed previously in Section 2.3.

Applications that incorporate aspect rely on the ability to first automatically identify the aspectual category of a clause. For example, Passonneau (1998) describes an algorithm that depends on what is called **lexical aspect**, the aspectual information stored in the lexicon for each verb, and Dorr (1992) augments Jackendoff's lexical entries with aspectual information. Combining linguistic indicators with machine learning automatically produces domain-specialized aspectual lexicons.

3. Linguistic Indicators

Aspectually categorizing verbs is the first step towards aspectually classifying clauses, since many clauses in certain domains can be categorized based on their main verb only (Siegel 1997, 1998b, 1999). However, the most frequent category of a verb is often domain dependent, so it is necessary to perform a specialized analysis for each domain.

3 These example sentences are modifications of samples from the corpus of medical discharge summaries described below.

Table 4
Fourteen linguistic indicators evaluated for aspectual classification.

Linguistic Indicator	Example Clause
`frequency`	(not applicable)
not or *never*	*She can* **not** *explain why.*
temporal adverb	*I saw to it* **then**.
no subject	*He was admitted to the hospital.*
past/pres participle	*. . . blood pressure going up.*
duration *in*-PP	*She built it* **in an hour**.
perfect	*They* **have** *landed.*
present tense	*I am happy.*
progressive	*I am behaving myself.*
manner adverb	*She studied* **diligently**.
evaluation adverb	*They performed* **horribly**.
past tense	*I was happy.*
duration *for*-PP	*I sang* **for ten minutes**.
continuous adverb	*She will live* **indefinitely**.

Naturally occurring text contains vast amounts of information pertaining to aspectual classification encoded by aspectual markers that have associated aspectual constraints. However, the best way to make use of these markers is not obvious. In general, while the presence of a marker in a particular clause indicates a constraint on the aspectual class of the clause, the absence thereof does not place any constraint.

Therefore, as with most statistical methods for natural language, the linguistic constraints associated with markers are best exploited by a system that measures co-occurrence frequencies. In particular, we measure the frequencies of aspectual markers across verbs. This way, the aspectual **tendencies** of verbs are measured. These tendencies are likely to correlate with aspectual class (Klavans and Chodorow 1992). For example, a verb that appears more frequently in the progressive is more likely to describe an event. The co-occurrence frequency of a linguistic marker is a linguistic indicator.

The first column of Table 4 lists the 14 linguistic indicators evaluated to classify verbs. Each indicator has a unique value for each verb. The first indicator, `frequency`, is simply the frequency with which each verb occurs over the entire corpus. The remaining 13 indicators measure how frequently each verb occurs in a clause with a linguistic marker listed in Table 4. For example, the next three indicators listed measure the frequency with which verbs (1) are modified by *not* or *never*, (2) are modified by a temporal adverb such as *then* or *frequently*, and (3) have no deep subject (e.g., passivized phrases such as, *She was admitted to the hospital*).

Nine of these indicators measure the frequencies of aspectual markers, each of which have linguistic constraints: perfect, progressive, duration *in*-PP, duration *for*-PP, no subject, and four adverb groups. The remaining five indicators were discovered during the course of this research. Further details regarding the measurement of these indicators, and the linguistic constraints that motivate them, can be found in Siegel (1998b).

Linguistic indicators are measured over corpora automatically. To do this, the automatic identification of individual constituents within a clause is required to detect the presence of aspectual markers and to identify the main verb of each clause. We employ the English Slot Grammar (ESG) parser (McCord 1990), which has previously been used on corpora to accumulate aspectual data (Klavans and Chodorow 1992). ESG is particularly attractive for this task since its output describes a clause's deep roles, detecting, for example, the deep subject and object of a passivized phrase.

4. Combining Linguistic Indicators with Machine Learning

There are several reasons to expect superior classification performance when employing multiple linguistic indicators in combination rather than using them individually. While individual indicators have predictive value, they are predictively incomplete. This incompleteness has been illustrated empirically by showing that some indicators help for only a subset of verbs (Siegel 1998b). Such incompleteness is due in part to sparsity and noise of data when computing indicator values over a corpus with limited size and some parsing errors. However, this incompleteness is also a consequence of the linguistic characteristics of various indicators. For example:

- While the progressive indicator is linguistically linked to extendedness, it is only indirectly linked to completedness. It may be useful for predicting whether a verb is culminated or nonculminated due to the fact that nonextended (i.e., atomic) verbs are more likely to be culminated than extended, i.e., points are rare.
- Many **location** verbs can appear in the progressive, even in their stative sense, e.g., *The book was lying on the shelf*.
- Some aspectual markers such as the pseudocleft and many manner adverbs test for *intentional* events in particular (not all events in general), and therefore are not compatible with all events, e.g., **I died* **diligently**.
- Aspectual coercion such as iteration can allow a punctual event to appear in the progressive, e.g. *She was sneezing for a week* (point → process → culminated process)[4] (Moens and Steedman 1988).
- The predictive power of some indicators is uncertain, since several measure phenomena that are not linguistically constrained by any aspectual category, e.g., the present tense, durative *for*-PPs, `frequency` and *not*/*never* indicators.

Therefore, the predictive power of individual linguistic indicators is incomplete; only the subset of verbs that adhere to the respective constraints or trends can be correctly classified. Such incomplete indicators may complement one another when placed in combination. Our goal is to take full advantage of the complete range of indicators by coordinating and combining them.

Machine learning methods can be employed to automatically generate a model that will combine indicator values. Figure 1 shows a system overview for this process (with additional details that are addressed below in Section 5.1.1). This diagram outlines a generic system that combines numerical indicators with machine learning for a classification problem, in natural language understanding or otherwise. Indicators are computed over an automatically parsed corpus. Then, in the Combine Indicators stage, supervised training cases are used to automatically generate a model (Classification Method) with supervised machine learning. This method (the hypothesis) inputs indicator values and outputs the aspectual class. The hypothesis is then evaluated over unseen supervised test cases.

4 In this example, *for a week* requires an extended event, thus the first coercion. However, this phrase also makes an event culminated, thus the second transformation.

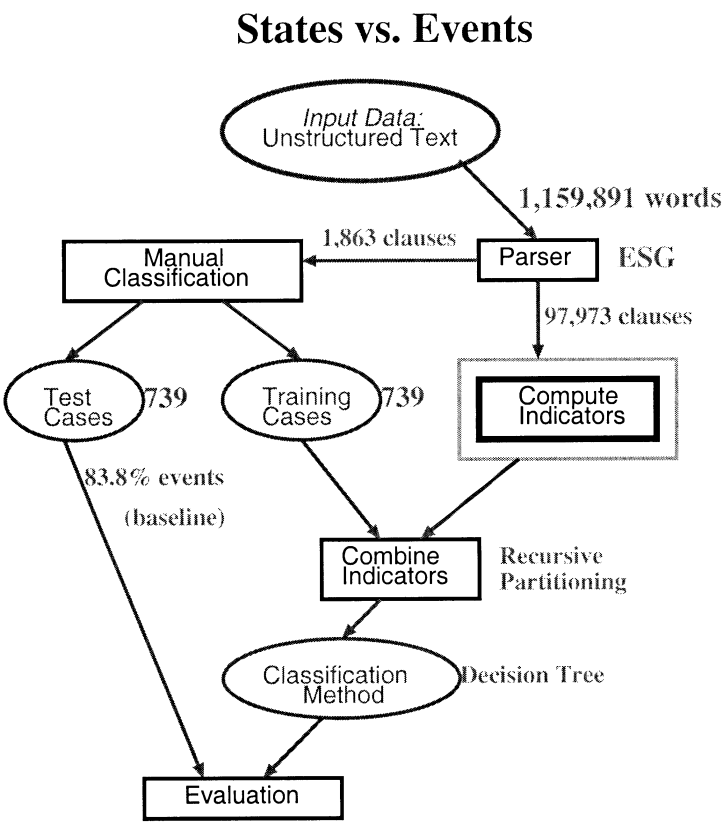


Figure 1
System overview with statistics of the medical discharge summary data for distinguishing according to stativity.

There are five advantages to automating this process with machine learning:

- The cost of *manually* generating a model, which is often prohibitive, is avoided.
- Biases introduced by a human engineer are avoided.
- Automated approaches are extensible to multiple natural language classification problems, across multiple domains and multiple languages.
- Once a system has been trained to distinguish verbs by indicator values, it can automatically classify all the verbs that appear in a corpus, including unseen verbs that were not included in the supervised training sample.
- Resulting models may reveal new linguistic insights.

The remainder of this section describes the three supervised learning methods evaluated for combining linguistic indicators: logistic regression, decision tree induction, and a genetic algorithm. At the end of this section, the designs of these three methods are compared. In the following section, the three are compared empirically: each method is evaluated for classification according to both stativity and completedness.

4.1 Logistic Regression

As suggested by Klavans and Chodorow (1992), a weighted sum of multiple indicators that results in one "overall" indicator may provide an increase in classification performance. This method follows the intuition that each indicator correlates with the probability that a verb belongs in a certain class, but that each indicator has its own unique scale, polarity, and predictive significance and so must be weighted accordingly.

For example, consider the problem of using in combination (only) two indicators, *not/never* and progressive, to determine the stativity of verbs in a corpus of medical reports. The former indicator may show higher values for stative verbs since diagnoses (i.e., states) are often ruled out in medical discharge summaries, e.g., *"The patient was* **not** *hypertensive,"* but procedures (i.e., events) that were not done are not usually mentioned, e.g., *"***?***An examination was* **not** *performed."* The progressive indicator is linguistically predicted to show higher values for event verbs in general, so its polarity is the opposite of the *not/never* indicator. Furthermore, a certain group of stative verbs (including, e.g., *sit*, *lay*, and *rest*) can also occur in the progressive, so this indicator may be less powerful in its predictiveness. Therefore, the best overall indicator may result from adding the value of the *not* or *never* indicator, as multiplied by a negative weight, to the value of the progressive indicator, as multiplied by a positive weight of lesser magnitude. A detailed examination of the weights resulting from learning and their linguistic interpretation is described below in Section 5.1.4.

This set of weights can be determined by a standard gradient descent algorithm (see, for example, Mitchell [1997]). However, the algorithm employed here is logistic regression (Sjantner and Duffy 1989), a popular technique for binary classification. This technique determines a set of weights for a linear model, which are applied in combination with a certain nonlinear model. In particular, the iterative reweighted least squares algorithm (Baker and Nelder 1989) is employed, and the inverse logic (nonlinear) function is applied. The Splus statistical package was used for the induction process.

4.2 Decision Tree Induction

Another method capable of modeling nonlinear relationships between indicators is a decision tree. An example tree is shown in Figure 2 (with additional details discussed in Section 5.1.4). Each internal node of a decision tree is a choice point, dividing an individual indicator into two ranges of possible values by way of a threshold. Each leaf node is labeled with a classification (e.g., state or event, in the case of the tree shown). In effect, this is simply a set of nested *if-then-else* statements. Given the set of indicator values corresponding to a verb, that verb's class is predicted by traversing the tree from the root node to a leaf as follows: at each node, the arc leading downward to the left or right is traversed according to the question posed about an indicator value at that node. When a leaf node is reached, its label is then taken to be the verb's classification. For example, if the `frequency` is less than 2,013, the arc to the left is traversed. Then, if the *not/never* indicator is greater than or equal to 3.48%, the arc to the right is traversed. Finally, if the `frequency` is greater than or equal to 314, the arc to the right is traversed, arriving at a leaf labeled, State.

This representation enables complex interactions between indicator values. In particular, if one indicator can only help classify a proper subset of verbs, while another applies only to a subset that is distinct but intersects with the first, certain ranges of indicator values may delimit verb groups for which the indicators complement one another. An example of such delimitation within a learned decision tree is illustrated below in Section 5.1.4.

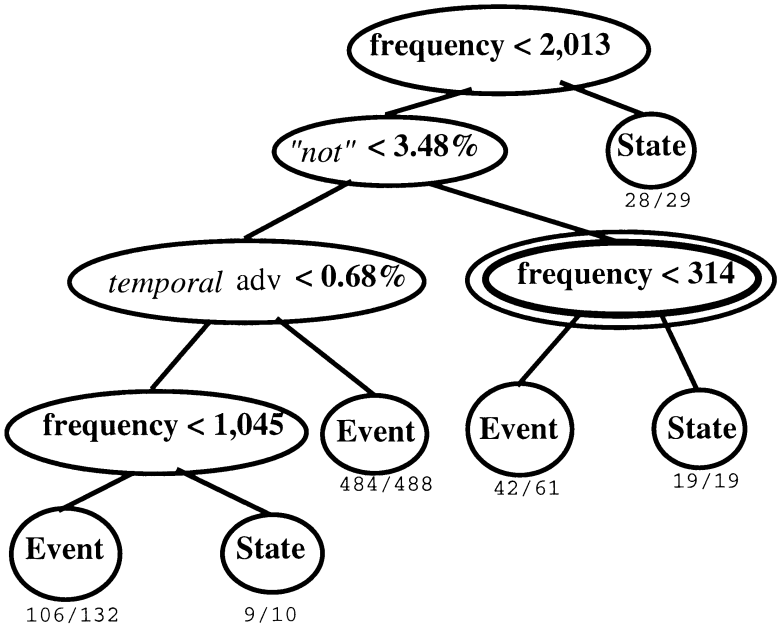


Figure 2
Top portion of decision tree automatically created to distinguish events from states. Leftward arcs are traversed when comparisons test true, rightward arcs when they test false. The values under each leaf indicate the number of correctly classified examples in the corresponding partition of training cases. The full tree has 59 nodes and achieves 93.9% accuracy over unseen test cases.

Table 5
Default decision tree induction parameters implemented by Splus.

Minimum partition size before first split:	5
Minimum partition size for additional splits:	10
Node selection criterion:	deviance = −2 times log-likelihood
Node purity threshold:	deviance < .01

The most popular method of decision tree induction, which we employ here, is recursive partitioning (Quinlan 1986; Breiman et al. 1984). This method "grows" a decision tree by expanding it from top to bottom. Initially, the tree has only one node, a leaf, corresponding to the entire set of training examples. Then, the following process is repeated: At each leaf node of the tree, an indicator and threshold are selected such that the examples are best distinguished according to aspectual class. This adds two leaves beneath the node, and distributes the examples to the new leaves accordingly.

Table 5 shows the parameters used to control decision tree growth. The criterion optimized for each split is deviance, implemented as minus twice the log likelihood. The Splus statistical package was used for the induction process. We also compared these results to standard CART decision tree induction (Friedman 1977; Breiman et al. 1984).

4.3 Genetic Programming

An alternative method that enables arbitrary mathematical combinations of indicators is to generate function trees that combine them. A popular method for generating

Table 6
Applying genetic programming to induce a function tree that combines linguistic indicators.

Objective:	Function tree to combine linguistic indicators
Terminal set:	14, corresponding to the set of linguistic indicators.
Function set:	ADD, MULTIPLY, and DIVIDE
Training cases:	739 (stativity) or 307 (completedness) verb instances.
Raw fitness:	Accuracy over training cases, when best threshold is selected.
Parameters:	Number of generates = 50,000, population size = 500, 5-member tournament selection, steady state population (Syswerda 1989).
Identification of best of run:	Highest raw fitness.

such function trees is a genetic algorithm (GA) (Holland 1975; Goldberg 1989), which is modeled after population genetics and natural selection. The use of GAs to generate function trees (Cramer 1985; Koza 1992) is often called genetic programming (GP).

Inspired by Darwinian survival of the fittest, the GA works with a pool of initially random hypotheses (in this case, function trees), stochastically performing genetic recombination and mutation to propagate or create better individuals, which replace old or less good individuals. Recombination between function trees usually consists of selecting a subtree from each individual, and swapping them, thereby creating two new individuals (Cramer 1985; Koza 1992). For random mutation, a randomly chosen subtree can be replaced by a new randomly created subtree (Koza 1992). Because the genetic algorithm is stochastic, each run may produce a different function tree. Therefore, performance is evaluated over the models produced by multiple runs.

The function trees are generated from a set of 17 primitives: the binary functions ADD, MULTIPLY, and DIVIDE, and 14 terminals corresponding to the 14 indicators listed in Table 4, which are occurrence frequencies. The GA parameters are shown in Table 6.

This representation enables strategic combinations of indicator values that are mathematical, as opposed to logical, manipulations. For example, two indicators that are high in predictiveness can be multiplied together, and perhaps added to an indicator with complementary but less reliable predictiveness.

The set of primitives was established empirically; other primitives such as conditional functions, subtraction, and random constants failed to improve performance. The polarities for several indicators were reversed according to the polarities of the weights established by logistic regression for stativity. Runs of the GA maintain a population size of 500 and end after 50,000 new individuals have been evaluated.

A threshold must be selected for both logistic and function tree combinations of indicators so overall outputs can be discriminated to maximize classification performance. For both methods, the threshold is established over the training set and frozen for evaluation over the test set.

4.4 Selecting and Comparing Learning Methods

The use of machine learning techniques to combine numerical indicators for classification problems in general is a well-established practice and includes work with decision trees (Fayyad and Irani 1992), logistic regression (Sjantner and Duffy 1989), and GP (Koza 1992; Tackett and Carmi 1994). Applications include document classification (Masand 1994), image classification (Tackett 1993), and stock market prediction (Allen and Karjalainen 1995).

When combining linguistic indicators in particular, the choice of hypothesis representation determines the type of linguistic insights that can result. Decision trees can be analyzed by examining the subset of verbs that are sorted to a particular node and the constraints on indicator values that put them there. A path from the root to any node is a rule that puts constraints on indicator values; this rule can be examined to determine if it has a linguistic interpretation. The weights produced by logistic regression can be examined to see which indicators are most highly weighted for each classification task. In addition to this, surprisingly, we discovered a decision tree–like rule encoded by these weights, as described below in Section 5.1.4. On the other hand, a function tree, such as that produced by GP, is more difficult to analyze manually, since it is a mathematical combination. However, GP's performance was tested due to the potential improvement in classification performance of such a flexible representation for numerical functions.

The relative merit of various learning algorithms is often difficult to ascertain, even after results have been collected. In general, each learning algorithm relies on an inductive bias, that may produce better results for some data than for others (Mitchell 1997). When applied to linguistic indicators, there is little knowledge about how indicators interact, since initial analysis examines individual indicators in isolation; machine learning is being used to automatically discover their interaction. Intuition guides the choice and design of algorithms, such as the rationale for each of the three techniques described above in this section. Moreover, beyond the particular characteristics of any given classification task, the particular data sample to which a learning technique is applied may have a large effect on performance, for example, due to the distribution and size of the training set, differences between the distributions of the training and test sets, and even the particular order in which the training cases are listed.

The three learning methods we examine in detail are diverse in their representation abilities, as described in this section, and, arguably, are therefore representative of the abilities of learning algorithms in general when applied to the same data. A pilot study showed no further improvement in accuracy or in recall trade-off for either classification problem by another four standard learning algorithms: naive Bayes (Duda and Hart 1973), Ripper (Cohen 1995), ID3 (Quinlan 1986) and C4.5 (Quinlan 1993). Furthermore, using metalearning to combine multiple learning methods hierachically (sometimes called stacked generalization; Chan and Stolfo [1993], and Wolpert [1992]), according to the JAM (Java Agents for Metalearning) model (Stolfo et al. 1997), produced equivalent results. However, this may be due to the limited size of our supervised data.

5. Supervised Learning: Method and Results

In this section, we evaluate the models produced by the three supervised learning methods. These methods are applied to combine the linguistic indicators computed over the medical discharge summaries in order to distinguish between states and events. Then, the methods are applied to indicators computed over novels in order to distinguish between culminated and nonculminated events. At the end of this section, these results are compared to one another, and to an informed baseline classification method.

The two data sets are summarized in Table 7. Table 8 illustrates the schema of inputs for supervised learning. There are a total of 14 continuous attributes for the two binary learning problems. All attributes are proportions except `frequency`, which is a positive integer. This data is available at http://www.cs.columbia.edu/~evs/VerbData/.

Table 7
Two classification problems on different data sets.

	Stativity	**Completedness**
Corpus	3,224 medical discharge summaries	10 novels
Corpus size	1,159,891 words	846,913 words
Parsed clauses	97,973	75,289
Training clauses	739 (634 events)	307 (196 culminated)
Testing clauses	739 (619 events)	308 (195 culminated)
Verbs in test set	222	204
Clauses excluded	*be-*, *have*-clauses	stative clauses
Unsupervised results	N/A	stativity (see Section 6)

Table 8
Schema of inputs for supervised learning. Fourteen continuous attributes for two binary learning problems: stativity and completedness.

Linguistic Indicator	**stativity**	**completedness**
Class	**yes..no**	**yes..no**
manner adverb	0.00..0.29	0.00..0.23
duration *in*-PP	0.00..0.13	0.00..0.12
continuous adverb	0.00..1.00	0.00..0.33
temporal adverb	0.00..0.40	0.00..0.97
not or *never*	0.47..1.00	0.58..1.00
duration *for*-PP	0.00..0.15	0.00..1.00
perfect	0.00..0.35	N/A
perfect (not progressive)	N/A	0.00..0.50
past/pres participle	0.00..1.00	0.33..1.00
evaluation adverb	0.00..0.46	0.00..0.95
no subject	0.00..1.00	0.00..1.00
past tense	0.00..1.00	0.00..1.00
present tense	0.00..1.00	0.00..1.00
`frequency`	1..2,131	1..13,882
not progressive	0.00..1.00	0.00..0.50

For both classification problems, we show that individual indicators correlate with aspectual class, but attain limited classification accuracy when used alone. Supervised learning is then used to combine indicators, improving classification performance and providing linguistic insights. The results of unsupervised learning are given in Section 6.

Classification performance is evaluated according to a variety of performance measures, since some applications weigh certain classes more heavily than others (Brodley 1996; Cardie and Howe 1997). An alternative to evaluation based on overall accuracy is to measure the recall values for the dominant and nondominant (i.e., majority and minority) categories separately. A favorable recall trade-off is achieved if the recall of the nondominant category can be improved with no loss in overall accuracy when compared against some baseline (Cardie and Howe 1997). Achieving such a trade-off is nontrivial; it is not possible, for example, for an uninformed approach that assigns everything to the dominant category. A favorable recall trade-off presents an advantage for applications that weigh the identification of nondominant instances, e.g., nonculminated clauses, more heavily. For example, correctly identifying the use of *for* depends on identifying the aspectual class of the clause it modifies (see Section 2.3). A system that summarizes the duration of events which incorrectly classifies *She ran (for*

a minute) as culminated will not detect that *for a minute* describes the duration of the *run* event. As another example, it is advantageous for a medical system that retrieves patient diagnoses to identify stative clauses, since there is a correspondence between states and medical diagnoses.

Classification performance is evaluated over verb instances, that is, clauses in which the verb appears as the main verb.[5] Because of this, as discussed further in Section 5.4 below, the same verb may appear multiple times in the training and testing sets. This measure is beneficial in several ways:

- Measured classification performance reflects the true distribution of the verbs—some are more frequent than others.
- Ambiguous verbs may appear with multiple aspectual categories, reflecting the true distribution of the data.
- In related work, clausal constituents other than the verb can be incorporated to help resolve ambiguity and alleviate verb sparsity (Siegel 1998a, 1998b).

5.1 States versus Events

Our experiments distinguishing states and events were performed across a corpus of 3,224 medical discharge summaries, with a total of 1,159,891 words. A medical discharge summary describes the symptoms, history, diagnosis, treatment, and outcome of a patient's visit to the hospital. Each summary consists of unstructured text, divided into several sections with titles such as: "History of Present Illness," and "Medical Summary." The text under four of these titles was extracted and parsed with the English Slot Grammar, resulting in 97,973 clauses that were parsed fully, with no self-diagnostic errors (ESG produced error messages on some of this corpus' complex sentences). Other sections in the summaries were ignored since they report the numerical results of certain medical tests, interspersed with incomplete sentences.

Be and *have*, the two most popular verbs, covering 31.9% of the clauses in this corpus, are handled separately from all other verbs. Clauses with *be* as their main verb, composing 23.9% of the corpus, always denote a state. Therefore, we can focus our efforts on the remaining clauses. Clauses with *have* as their main verb, composing 8.0% of the corpus, are highly ambiguous, and have been addressed separately by considering the direct object of such clauses (Siegel 1998a, 1998b).

5.1.1 Manual Marking for Supervised Data. As a basis for evaluating our approach, 1,851 clauses from the parsed corpus were manually marked according to their fundamental stativity. In contrast to the entire parsed corpus (97,973 clauses), each clause in this set of supervised data had to be judged by a linguist. This subset was selected uniformly from clauses in the corpus that had main verbs other than *be* and *have*. As a linguistic test for marking, each clause was tested for readability with *What happened was* .... Manual labeling followed a strict set of linguistically motivated guidelines in order to ascertain fundamental aspectual class. Modifiers that result in aspectual transformations, such as durative *for*-PPs, and in exceptions, such as *not*, were ignored. A comparison between human markers for this test was performed over a different corpus, and is reported below in Section 5.2.1.

5 For evaluation over sets of unique verbs, see Siegel (1998b).

Table 9
Indicators discriminate between states and events.

Linguistic Indicator	Stative Mean	Event Mean	T-test P-value
`frequency`	`932.89`	`667.57`	0.0000
not or *never*	4.44%	1.56%	0.0000
temporal adverb	1.00%	2.70%	0.0000
no subject	36.05%	57.56%	0.0000
past/pres participle	20.98%	15.37%	0.0005
duration *in*-PP	0.16%	0.60%	0.0018
perfect	2.27%	3.44%	0.0054
present tense	11.19%	8.94%	0.0901
progressive	1.79%	2.69%	0.0903
manner adverb	0.00%	0.03%	0.1681
evaluation adverb	0.69%	1.19%	0.1766
past tense	62.85%	65.69%	0.2314
duration *for*-PP	0.59%	0.61%	0.8402
continuous adverb	0.04%	0.03%	0.8438

Because of manually identified parsing problems (verb or direct object incorrectly identified), 373 clauses were rejected. This left 1,478 clauses, which were divided equally into training and testing sets of 739 clauses each.

Figure 1 (see Section 4) shows the system overview with details regarding the medical discharge summary corpus used in this study. As this shows, the values for linguistic indicators are computed across the entire parsed corpus. This is a fully automatic process. Then, the 739 training examples are used to derive a mechanism, e.g., a decision tree, for combining multiple indicators. The combination of indicators achieves an increase in classification performance. This increase in performance is then validated over the 739 unseen test cases.

5.1.2 Upper and Lower Bounds in Accuracy. Of clauses with main verbs other than *be* and *have*, 83.8% are events. Therefore, simply classifying every verb as an event achieves an accuracy of 83.8% over the 739 test cases. However, this approach classifies all state clauses incorrectly, achieving a **stative recall** of 0.0%. This method serves as a baseline for comparison, since we are attempting to improve over an uninformed approach.[6]

One limitation to our approach places an upper bound on classification accuracy. Our approach examines only the main verb, since linguistic indicators are computed for verbs only. For example, a verb occurring three times as an event and twice as a state will be misclassified at least two of the five times. This limits classification accuracy to a maximum of 97.4% over the test cases due to the presence of verbs with multiple classes. The ramifications of this limitation are explored below in Section 5.4.

5.1.3 Individual Indicators. The values of the 14 indicators listed in Table 9 were computed, for each verb, across the 97,973 parsed clauses from our corpus of medical discharge summaries. As described in Section 3, each indicator has a unique value for each verb, which corresponds to the frequency of the aspectual marker with the verb (except verb `frequency`, which is an absolute measure over the corpus).

6 Similar baselines for comparison have been used for many classification problems (Duda and Hart 1973), e.g., part-of-speech tagging (Church 1988; Allen 1995).

The second and third columns of Table 9 show the average value for each indicator over stative and event clauses, as measured over the training examples (which exclude *be* and *have*). These values are computed solely over the 739 training cases in order to avoid biasing the classification experiments in the sections below, which are evaluated over the unseen test cases. For example, for the *not/never* indicator, stative clauses have an average value of 4.44%, while event clauses have an average value of 1.56%. This makes sense, since diagnoses are often ruled out in medical discharge summaries, e.g., *The patient was* **not** *hypertensive*, but procedures that were not done are not usually mentioned, e.g., *?An examination was* **not** *performed*.

The differences in stative and event means are statistically significant ($p < .01$) for the first seven of the 14 indicators listed in Table 9. The fourth column shows the results of t-tests that compare indicator values over stative verbs to those over event verbs for each indicator. For example, there is less than a 0.05% chance that the differences between stative and event means for the first seven indicators listed is due to chance. The differences in average value for the bottom seven indicators were not confirmed to be significant with this small sample size (739 training examples).

A positive correlation between indicator value and verb class does not necessarily mean an indicator can be used to increase classification accuracy over the baseline of 83.8%. This is because of the dominance of events among the testing examples; a threshold to distinguish verbs that correctly classifies more than half of each class will have an accuracy lower than the baseline if the number of states correctly classified is less than the number of events misclassified. To examine this, each indicator was tested individually for its ability to improve classification accuracy over the baseline by establishing the best classification threshold over the training data. The performance of each indicator was validated over the testing data using the same threshold.

Only the `frequency` indicator succeeded in significantly improving classification accuracy. Both `frequency` and occurrences with *not* or *never* improved classification accuracy on the training data over the baseline obtained by classifying all clauses as events. To validate this improved accuracy, the thresholds established over the training set were used over the test set, with resulting accuracies of 88.0% and 84.0%, respectively. Binomial tests show the first of these, but not the second, to be a significant improvement over the baseline of 83.8%.

This improvement in accuracy was achieved simply by discriminating the popular verb *show* as a state, but classifying all other verbs as events. Although many domains may primarily use *show* as an event, in its appearances in medical discharge summaries, such as *His lumbar puncture* **showed** *evidence of white cells*, *show* primarily denotes a state. This observation illustrates the importance of empirical techniques for lexical knowledge acquisition.

5.1.4 Indicators Combined with Learning. All three machine learning methods successfully combined indicator values, improving classification accuracy over the baseline measure. As shown in Table 10, the decision tree's accuracy was 93.9%, GP's function trees had an average accuracy of 91.2% over seven runs, and logistic regression achieved an 86.7% accuracy (Baseline 2 is discussed below in Section 5.4). Binomial tests showed that both the decision tree and GP achieved a significant improvement over the 88.0% accuracy achieved by the `frequency` indicator alone. These results show that machine learning methods can successfully combine multiple numerical indicators to improve verb classification accuracy.

The increase in the number of stative clauses correctly classified, i.e., stative recall, illustrates an even greater improvement over the baseline. As shown in Table 10, stative

Table 10
Comparison of three learning methods, optimizing for accuracy, and two performance baselines, distinguishing states from events.

	Overall Accuracy	States		Events	
		Recall	Precision	Recall	Precision
Decision tree	93.9%	74.2%	86.4%	97.7%	95.1%
GP (7 runs)	91.2%	47.4%	97.3%	99.7%	90.7%
Logistic	86.7%	34.2%	68.3%	96.9%	88.4%
Baseline 1	83.8%	0.0%	100.0%	100.0%	83.8%
Baseline 2	94.5%	69.2%	95.4%	99.4%	94.3%

Table 11
Comparing training and test performance on three learning methods, distinguishing states from events.

	Training Accuracy	Testing Accuracy
Decision tree	96.3%	93.9%
GP	93.4%	91.2%
Logistic	88.8%	86.7%
Baseline	85.8%	83.8%

recalls of 74.2%, 47.4%, and 34.2% were achieved by the three learning methods, as compared to the 0.0% stative recall achieved by Baseline 1, while only a small loss in recall over event clauses was suffered. The baseline does not classify any stative clauses correctly because it classifies all clauses as events. This difference in recall is more dramatic than the accuracy improvement because of the dominance of event clauses in the test set.

Overfitting was moderate for each of the three supervised learning algorithms. As shown in Table 11, each learning method's performance over the training data was about two points higher than that over the test data. This corresponds to a two-point difference in baseline performance, which is due to a higher proportion of event clauses in the training data.

The thresholds established to discriminate the outputs of GP's function trees generalize well to unseen data. When inducing these function trees with the GA, the training set is used to form the tree, and to select a threshold that best discriminates between verbs of the different classes. There is the potential that a threshold determined over the training cases will not generalize well when evaluated over the test cases. To test this, for each of the seven function trees generated by the GA to distinguish between states and events, the best threshold was selected over the *test* cases. For five of the function trees, there was no threshold that increased classification accuracy beyond that attained by the threshold established over the training cases. For the other two, a threshold was found that allowed for *one* more of the 739 test cases to be correctly classified.

In the remainder of this section, we compare the resulting models of the three supervised learning method and contrast the ways in which they combine indicators.

Logistic Regression. Logistic regression successfully combined the 14 linguistic indicators, achieving an accuracy of 86.7%, as shown in Table 10. This is a significant improvement over the baseline accuracy of 83.8%, as measured with a binomial test. Furthermore, a stative recall of 34.2% was achieved.

Table 12
Weights produced by logistic regression to distinguish between stative and event verbs.

Linguistic Indicator	Logistic Weight	T-test P-value
manner adverb	11.04744	0.1681
duration *in*-PP	0.06209624	0.0018
continuous adverb	– 0.04168417	0.8438
temporal adverb	0.02127572	0.0000
not or *never*	0.01714499	0.0000
duration *for*-PP	– 0.01019155	0.8402
perfect	0.009528091	0.0054
past/pres participle	0.006981148	0.0005
evaluation adverb	0.005695407	0.1766
no subject	0.002742867	0.0000
past tense	0.002586572	0.2314
present tense	0.002409898	0.0901
`frequency`	– 0.001264895	0.0000
not progressive	– 0.0009369231	0.0903

The particular weighting scheme resulting from logistic regression for this data effectively integrates a decision tree type rule, along with the usual weighting of logistic regression. This is illustrated in Table 12, which shows the weights automatically derived by logistic regression for each of the 14 linguistic indicators. The value assigned to the manner adverb indicator, 11.04744, far outweighs the other 13 weights. At first glance, it may appear that this weighting scheme favors the manner adverb indicator over all other indicators. However, as shown in Table 13, manner adverb indicator values are 0.0% for all verbs in the training set except the eight indicated, all of which denote events. Therefore, the large weight assigned to the manner adverb indicator is only activated for those verbs, which are therefore each classified as events, regardless of the remaining 13 indicator values. For all other verbs, the remaining 13 indicator values determine the classification.

This rule cannot increase accuracy over the baseline without the remaining 13 indicators, since it does not positively identify any states—it only identifies events, which are all correctly classified by the baseline. Therefore, it is only useful because the overall model also correctly identifies some stative clauses.

Genetic Programming. GP successfully combined the 14 linguistic indicators, achieving an average accuracy of 91.2%, as shown in Table 10. This is a significant improvement over the baseline accuracy of 83.8%, according to a binomial test. Furthermore, a stative recall of 47.4% was achieved.

GP improved classification performance by emphasizing a different set of indicators than those emphasized by logistic modeling. Figure 3 shows an example function tree automatically generated by the GA, which achieved 92.7% accuracy. Note that this classification performance was attained with a subset of only five linguistic indicators: duration *in*-PP, progressive, *not* or *never*, past tense, and `frequency`. Two of these are ranked lowest by logistic regression: `frequency` and progressive. Furthermore, manner adverb, ranked highest by logistic regression, is not incorporated in this function tree at all. This may be because this indicator only applies to a small number of verbs, as shown in Table 13, and because an *if*-rule such as that captured by logistic regression is difficult to encode with a function tree with no conditional functions. Overall, we can conclude that multiple proper subsets of linguistic indicators are useful for aspectual classification if combined with the correct model.

```
(/ (+ (+ (* (/ DurationInPP (+ Progressive (+ NotNever NotNever))) (+
Progressive 75)) (/ (* (+ Progressive PastTense) Progressive)
NegFrequency)) NotNever) DurationInPP)
```

Figure 3
Example function tree designed by a genetic algorithm to distinguish between stative and event verbs, achieving 92.7% accuracy.

Table 13
Linguistic rule discovered by logistic regression.

If frequency with manner adverbs is ...	verbs this applies to:	Frequency in Training Set	then classify as:
> 0.0%	*adjust*	1	Event
	continue	16	
	decline	2	
	decrease	4	
	improve	5	
	increase	4	
	progress	2	
	resolve	7	
0.0%	all other verbs	699	depending on other 13 indicators

Decision Tree Induction. Decision tree induction successfully combined the 14 linguistic indicators, achieving the highest accuracy of the three supervised learning algorithms tested, 93.9%, as shown in Table 10. This is a significant improvement over the baseline accuracy of 83.8%, as measured with a binomial test. Furthermore, a stative recall of 74.2% was achieved. The top portion of the tree created with recursive partitioning is shown in Figure 2 (in Section 4.2, where it is explained). Note that the root node simply distinguishes the stative verb *show* with the `frequency` indicator, as described in Section 5.1.3.

To achieve this increase in classification performance, the decision tree divided the training cases into relatively small partitions of verbs. Table 14 shows the distribution of training case verbs examined by the highlighted tree node in Figure 2. As seen by tracing the path from the root to the highlighted node, these are the verbs with `frequency` less than 2,013 across the corpus, and modified by *not* or *never* at least 3.48% of the time. From this subset, the highlighted node distinguishes the three verbs with `frequency` at least 314, shown in capitals in Table 14, as states. This is correct for all 19 instances of these three verbs, and does not misclassify any event verbs.

This example illustrates a benefit of distinguishing verbs based on indicator values computed over large corpora. Most of the verbs in Table 14 appear in the training set a small number of times, so it would be difficult for a classification system to generate rules that apply to these individual *verbs*. Rather, since our system draws generalizations over the *indicator values* of verbs, it identifies stative verbs without misclassifying any of the event verbs shown.

Classification performance is equally competitive without the `frequency` indicator. Since `frequency` is the only indicator that can increase accuracy by itself, and since it is the first discriminator of the decision tree, it may appear that `frequency` highly dominates the set of indicators. This could be problematic, since the relationship between verb `frequency` and verb category may be particularly domain dependent, in which case `frequency` could be less informative when applied to other domains. However, when decision tree induction was employed to combine only the 13 indicators other

Table 14
Verb distribution in the partition of training examples sorted to the decision tree node highlighted in Figure 2. The three stative verbs shown in capitals are distinguished at this node, with no event verbs misclassified.

Events:							
get	3	talk	1	persue	1	drive	1
transfuse	2	suggest	1	provide	1	drink	1
respond	2	subside	1	pass	1	document	1
lose	2	sleep	1	notice	1	detect	1
live	2	retract	1	load	1	change	1
interfere	2	repair	1	limit	1	achieve	1
breathe	2	relieve	1	lift	1	regain	1
visualize	1	recognize	1	help	1		
tell	1	radiate	1	explain	1		
States:							
FEEL	12	associate	3	want	1	concern	1
know	4	remember	2	support	1	account	1
APPEAR	4	believe	2	indicate	1		
REQUIRE	3	accompany	2	desire	1		

Table 15
Indicators discriminate between culminated and nonculminated events.

Linguistic Indicator	Culminated Mean	Nonculminated Mean	T-test P-value
perfect	7.87%	2.88%	0.0000
temporal adverb	5.60%	3.41%	0.0000
manner adverb	0.19%	0.61%	0.0008
progressive	3.02%	5.03%	0.0031
past/pres participle	14.03%	17.98%	0.0080
no subject	30.77%	26.55%	0.0241
duration *in*-PP	0.27%	0.06%	0.0626
present tense	17.18%	14.29%	0.0757
duration *for*-PP	0.34%	0.49%	0.1756
continuous adverb	0.10%	0.49%	0.2563
`frequency`	`345.86`	`286.55`	0.5652
not or *never*	3.41%	3.15%	0.6164
evaluation adverb	0.46%	0.39%	0.7063
past tense	53.62%	54.36%	0.7132

than `frequency`, the resulting decision tree achieved 92.4% accuracy and 77.5% stative recall. Therefore, our results are not entirely dependent on the `frequency` indicator.

5.2 Culminated versus Nonculminated Events

In medical discharge summaries, nonculminated event clauses are rare. Therefore, our experiments for classification according to completedness are performed across a corpus of 10 novels, comprising 846,913 words. These novels were parsed with the English Slot Grammar, resulting in 75,289 clauses that were parsed fully, with no self-diagnostic errors. The values of the 14 indicators listed in Table 15 were computed, for each verb, across the parsed clauses. Note that in this section, the perfect indicator differs in that we ignore occurrences of the perfect in clauses that are also in the progressive, since any progressive clause can appear in the perfect, e.g., *I have been painting*.

5.2.1 Manual Marking for Supervised Data. To evaluate the performance of our system, we manually marked 884 clauses from the parsed corpus according to their aspectual class. These 884 were selected evenly across the corpus from parsed clauses that do not have *be* as the main verb, since we are testing a distinction between events. Of these, 109 were rejected because of manually identified parsing problems (verb or direct object incorrectly identified), and 160 were rejected because they described states. This left 615 event clauses over which to evaluate classification performance. The division into training and test sets was derived such that the distribution of classes was equal between the two sets. This precaution was taken because preliminary experiments indicated difficulty in demonstrating a significant increase in classification accuracy for completedness. This process resulted in 307 training cases (196 culminated) and 308 test cases (195 culminated). Since 63.3% of test cases are culminated events, simply classifying every clause as culminated achieves an accuracy of 63.3% over the 308 test cases (Baseline 1A). This method serves as a baseline for comparison.

We used linguistic tests that were selected for this task by Passonneau (1988) from the constraints and entailments listed in Tables 2 and 3. First, the clause was tested for stativity with *What happened was . . . .* Then, as an additional check, we tested with the following rule: if a clause can be read in a pseudocleft, it is an event, e.g., *What its parents did was run off*, versus **What we did was know what is on the other side*. Second, if a clause in the past progressive necessarily entails the past tense reading, the clause describes a nonculminated event. For example, *We were talking just like men* (nonculminated) entails that *We talked just like men*, but *The woman was building a house* (culminated) does not necessarily entail that *The woman built a house*. The guidelines described above in Section 5.1 were used in order to test for fundamental aspectual class.

Cross-checking between linguists shows high agreement. In particular, in a pilot study manually annotating 89 clauses from the corpus of novels, two linguists agreed 81 times (i.e., 91%). Informal analysis suggests the remaining disagreement could be further divided in half by a few simple refinements of the annotation protocol. Furthermore, of 57 clauses agreed to be events, 46 were annotated in agreement with respect to completedness.

The verb *say*, which is a frequent point, i.e., nonculminated and nonextended, poses a challenge for manual marking. Points are misclassified by the test for completedness described above since they are nonextended and therefore cannot be placed in the progressive. Therefore, *say*, which occurs nine times in the test set, was marked incorrectly as culminated. After some initial experimentation, we switched the class of each occurrence of *say* in our supervised data to nonculminated. This change to *say* made the class distribution slightly uneven between training and test data.

5.2.2 Individual Indicators. The second and third columns of Table 15 show the average value for each indicator over culminated and nonculminated event clauses, as measured over the training examples. For example, for the perfect tense indicator, culminated clauses have an average value of 7.87%, while nonculminated clauses have an average value of 2.88%. These values were computed solely over the 307 training cases in order to avoid biasing the classification experiments in the sections below, which are evaluated over the unseen cases.

The differences in culminated and nonculminated means are statistically significant ($p < .05$) for the first six of the 14 indicators listed in Table 15. The fourth column shows the results of t-tests that compare indicator values over culminated verbs to those over nonculminated verbs. For example, there is less than a 0.05% chance that the differences between culminated and nonculminated means for the first six indicators listed is due

Table 16
Comparison of four learning methods, optimized for accuracy, and three performance baselines distinguishing culminated from nonculminated events.

	Overall Accuracy	Culminated		Nonculminated	
		Recall	Precision	Recall	Precision
CART	74.0%	86.2%	76.0%	53.1%	69.0%
Logistic	70.5%	83.1%	73.6%	48.7%	62.5%
Logistic 2	67.2%	81.5%	71.0%	42.5%	57.1%
GP (4 runs)	68.6%	77.3%	74.2%	**53.6%**	57.8%
Decision tree	68.5%	86.2%	70.6%	38.1%	61.4%
Baseline 1A	63.3%	100.0%	63.3%	0.0%	100.0%
Baseline 1B	49.0%	46.4%	63.3%	**53.6%**	36.7%
Baseline 2	70.8%	94.9%	69.8%	29.2%	76.7%

to chance. The differences in average value for the bottom eight indicators were not confirmed to be significant with this small sample size (307 training examples).

For completedness, no individual indicator used in isolation was shown to significantly improve classification accuracy over the baseline.

5.2.3 Indicators Combined with Learning. When distinguishing according to completedness, both CART and logistic regression successfully combined indicator values, improving classification accuracy over the baseline measure. As shown in Table 16, classification accuracies were 74.0% and 70.5%, respectively. A binomial test showed each to be a significant improvement over the 63.3% accuracy achieved by Baseline 1A. Although the accuracies attained by GP and decision tree induction, 68.6% and 68.5% respectively, are also higher than that of Baseline 1A, based on a binomial test this is not significant. However, this may be due to our small test sample size.

The increase in the number of nonculminated clauses correctly classified, i.e., nonculminated recall, illustrates a greater improvement over the baseline. As detailed in Table 16, nonculminated recalls of 53.1%, 48.7%, 53.6%, and 38.1% were achieved by the learning methods, as compared to the 0.0% nonculminated recall achieved by Baseline 1A. Baseline 1A does not classify any nonculminated clauses correctly because it classifies all clauses as events. This difference in recall is more dramatic than the accuracy improvement because of the dominance of culminated clauses in the test set. Note that it is possible for an uninformed approach to achieve the same nonculminated recall as GP, 53.6%, by arbitrarily classifying 53.6% of all clauses as nonculminated, and the rest as culminated. However, as shown in Table 16, the average performance of such a method (Baseline 1B) loses in comparison to GP, for example, in overall accuracy (49.0%) and nonculminated precision (36.7%).

All three supervised learning methods highly prioritized the perfect indicator. The induced decision tree uses the perfect indicator as its first discriminator, logistic regression ranked the perfect indicator as fourth out of 14 (see Table 17), and one function tree created by GP includes the perfect indicator as one of five indicators used together to increase classification performance (see Figure 4). Furthermore, as shown in Table 15, the perfect indicator tied with the temporal adverb indicator as most highly correlated with completedness, according to t-test results. This is consistent with the fact that, as discussed in Section 2.1, the perfect indicator is strongly connected to completedness on a linguistic basis.

GP maintained classification performance while emphasizing a different set of indicators than those emphasized by logistic regression. Figure 4 shows an example

Table 17
Weights produced by logistic regression to distinguish between culminated and nonculminated verbs. Contrast with Table 12.

Linguistic Indicator	Logistic Weight	T-test P-value
duration *in*-PP	−0.1207664	0.0626
manner adverb	0.03808262	0.0008
evaluation adverb	0.03212381	0.7063
perfect	−0.02304221	0.0000
temporal adverb	−0.01643347	0.0000
not or *never*	−0.01212703	0.6164
not progressive	−0.01059269	0.0031
no subject	−0.006891114	0.0241
past/pres participle	−0.004127672	0.0080
past tense	0.002484739	0.7132
present tense	0.00218274	0.0757
continuous adverb	−0.001775534	0.2563
duration *for*-PP	0.0001747421	0.1756
`frequency`	−0.0000916167	0.5652

```
(+ (- (- (+ (/ NoSubject Frequency) TemporalAdv) (- 83 Perfect)) (/ (/
NoSubject Frequency) Frequency)) (- (+ NotProgressive NotProgressive) (-
60 Perfect)))
```

Figure 4
Example function tree designed by a genetic algorithm to distinguish between culminated and nonculminated verbs, achieving 69.2% accuracy and 62.8% nonculminated recall.

function tree automatically generated by GP, which achieved 69.2% accuracy. Note that, as for stativity, this classification performance was attained with a subset of only five linguistic indicators: no subject, `frequency`, temporal adverb, perfect, and not progressive. (However, only two of these appeared in the example function tree for stativity shown in Figure 2: `frequency` and progressive.) Since multiple proper subsets of indicators succeed in improving classification accuracy, this shows that some indicators are mutually correlated.

5.3 Comparing Learning Results Across Classification Tasks

As shown above, learning methods successfully produced models that were specialized for the classification task. In particular, the same set of 14 indicators were combined in different ways, successfully improving classification performance for both stativity and completedness, and revealing linguistic insights for each.

However, it is difficult to determine which learning method is the best for verb classification in general, since their relative ranks differ across classification task and evaluation criteria. The relative accuracies of the three supervised learning procedures rank in opposite orders when comparing the results in classification according to stativity (Table 10) to results in classification according to completedness (Table 16). Furthermore, when measuring classification performance as the recall of the nondominant class (stative and nonculminated, respectively), the rankings are also conflicting when comparing results for the two classification tasks. The difficulties in drawing conclusions about the relative performance of learning techniques are discussed in Section 4.4.

The same two linguistic indicators were ranked in the top two positions for both aspectual distinctions by logistic regression. As shown in Tables 17 and 12, which give the weights automatically derived by logistic regression for each of the 14 linguistic

indicators, the manner adverb and duration *in*-PP indicators are in the top two slots for both weighting schemes, corresponding to the two aspectual distinctions. This may indicate that these two indicators are universally useful for aspectual classification, at least when modeling with logistic regression. However, the remaining rankings of linguistic indicators differ greatly between the two weighting schemes.

5.4 Indicators versus Memorizing Verb Aspect

In this work, clauses are classified by their main verb only. Therefore, disambiguating between multiple aspectual senses of the same verb is not possible, since other parts of the clause (e.g., verb arguments) are not available as a source of context with which to disambiguate. Thus, the improvement in accuracy attained reveals the extent to which, across the corpora examined, most verbs are dominated by one sense.

A competing baseline approach would be to simply memorize the most frequent aspectual category of each verb in the training set, and classify verbs in the test set accordingly. In this case, test verbs that did not appear in the training set would be classified according to majority class. However, classifying verbs and clauses according to numerical indicators has several important advantages over this baseline:

- **Handles rare or unlabeled verbs.** The results we have shown serve to estimate classification performance over unseen verbs that were not included in the supervised training sample. Once the system has been trained to distinguish by indicator values, it can automatically classify any verb that appears in unlabeled corpora, since measuring linguistic indicators for a verb is fully automatic. This also applies to verbs that are underrepresented in the training set. For example, as discussed in Section 5.1.4, one node of the resulting decision tree trained to distinguish according to stativity identifies 19 stative test cases without misclassifying any of 27 event test cases with verbs that occur only one time each in the training set.
- **Success when training doesn't include test verbs.** To test this, all test verbs were eliminated from the training set, and logistic regression was trained over this smaller set to distinguish according to completedness. The result is shown in Table 16 (logistic 2). Accuracy remained higher than Baseline 1A (Baseline 2 is not applicable), and the recall trade-off is felicitous.
- **Improved performance.** Memorizing majority aspect does not achieve as high an accuracy as the linguistic indicators for completedness, nor does it achieve as wide a recall trade-off for both stativity and completedness. These results are indicated as the second baselines (Baseline 2) in Tables 10 and 16, respectively.
- **Classifiers output scalar values.** This allows the trade-off between recall and precision to be selected for particular applications by selecting the classification threshold. For example, in a separate study, optimizing for F-measure resulted in a more dramatic trade-off in recall values as compared to those attained when optimizing for accuracy (Siegel 1998b). Moreover, such scalar values can provide input to systems that perform reasoning on fuzzy or uncertainty knowledge.
- **Expandable framework.** One form of expansion is that additional indicators can be integrated by measuring the frequencies of additional

aspectual markers. Furthermore, indicators measured over multiple clausal constituents (e.g., main verb-object pairs) alleviate verb ambiguity and sparsity and improve classification performance (Siegel 1998b).

- **Manual analysis reveals linguistic insights.** As summarized below in Section 9, our analysis reveals linguistic insights that can be used to inform future work.

6. Unsupervised Learning

Unsupervised methods for clustering words have been developed that do not require manually marked examples (Hatzivassiloglou and McKeown 1993; Schütze 1992). These methods automatically determine the number of groups and the number of verbs in each group.

This section evaluates an approach to clustering verbs developed and implemented by Hatzivassiloglou, based on previous work for semantically clustering adjectives (Hatzivassiloglou and McKeown 1993; Hatzivassiloglou 1997). This system automatically places verbs into semantically related groups based on the distribution of co-occurring direct objects. Such a system avoids the need for a set of manually marked examples for the training process. Manual marking is time consuming and domain dependent, requires linguistic expertise, and must be repeated on a corpus representing each new domain.

The clustering approach differs from our approach combining linguistic indicators in two significant ways. First, the method semantically groups words in a general sense—it is not designed or intended to group words according to any particular semantic distinction such as stativity or completedness. Second, this method measures a co-occurrence relation not embodied by any of the 14 linguistic indicators presented in this article: the direct object. Note, however, that there are several advantages to linguistic indicators that measure the frequency of linguistic phenomena such as the progressive over measuring the frequencies of open-class words (Siegel 1998b).

The clustering algorithm was evaluated over the corpus of novels, which, as shown in Table 7, has 75,289 parsed clauses. Clauses were eliminated from this set if they had no direct object, or if the direct object was a clause, a proper noun, or a pronoun, or was misspelled. This left 14,038 distinct verb-object pairs of varying frequencies.

Because the direct object is an open-class category (noun), occurrences of any particular verb-object pair are sparse as compared to the markers measured by the linguistic indicators. For example, *make dinner* occurs only once among the parsed clauses from the corpus of novels, but *make* occurs 34 times in the progressive. For this reason, the clustering algorithm was evaluated over a set of frequent verbs only: 56 verbs occurred more than 50 times each in the set of verb-object pairs. Of these, the 19 shown in Figure 5 were selected as an evaluation set because of the natural semantic groups they fall into. The groupings shown, which do not pay heed to aspectual classification in particular, were established manually, but are not used by the automatic grouping algorithm.

Figure 6 shows the output of the unsupervised system. Seven groups were created, each with two to four verbs. The grouping algorithm used by this system is designed for data that is not as sparse with respect to the frequencies of verb-object pairs, e.g., data from a larger corpus. Thus, this partition is not representative of the full power of the approach, and a larger amount of data could improve it significantly. For more detail on the clustering algorithm and further results see Hatzivassiloglou and McKeown (1993) and Hatzivassiloglou (1997).

```
1. sell(27) buy(20) acquire(8)

2. push(28) pull(45)

3. raise(68) lower(24)

4. leave(160) enter(78)

5. know(164) forget(50) learn(51)

6. love(18) hate(16) like(112)

7. want(60) need(69) require(57) demand(15)
```

Figure 5
The set of verbs manually selected for evaluating unsupervised clustering, with frequencies shown. The grouping shown here was established manually.

```
1. *hate *like pull

2. lower raise

3. demand *know *love *want

4. buy sell

5. enter forget learn

6. acquire *need *require

7. leave push
```

Figure 6
Verb groupings created automatically by an unsupervised learning algorithm developed and implemented by Hatzivassiloglou and McKeown (1993) and Hatzivassiloglou (1997) applied over the corpus of 10 novels. Stative verbs are shown with an asterisk, and event verbs without.

The algorithm clearly discriminated event verbs from stative verbs.[7] In Figure 6, stative verbs are shown with an asterisk; event verbs are shown without. Three of the groups are dominated by stative verbs, and the other four groups are composed entirely of event verbs. Each stative verb is found in a group with 70.2% states, averaged across the 7 stative verbs, and each event verb is found in a group with 82.6% events, averaged across the 12 event verbs. This is an improvement over an uninformed baseline system that randomly creates groups of two or more verbs each, which would achieve average precisions of 63.2% and 36.8%, respectively.

We can draw two important conclusions from this result. First, unsupervised learning is a viable approach for classifying verbs according to particular semantic distinctions such as stativity. Second, co-occurrence distributions between the verb and direct

7 The algorithm also grouped verbs according to semantic relatedness in general, as can be seen by comparing the manual and automatic groupings. Further analysis of such results are given by Hatzivassiloglou and McKeown (1993).

object inform the aspectual classification of verbs. This is an additional source of information beyond the 14 linguistic indicators we combine with supervised learning.

7. Related Work

The aspectual classification of a clause has thus far been primarily approached from a knowledge-based perspective. For example, Pustejovsky's generative lexicon describes semantic interactions between clausal constituents that effect aspectual class (Pustejovsky 1991). Additionally, Resnik (1996) demonstrates the influence of implicit direct objects on aspectual classification.

The application of automatic corpus-based techniques to aspectual classification is in its infancy. Klavans and Chodorow (1992) pioneered the application of statistical corpus analysis to aspectual classification by placing verbs on a scale according to the frequency with which they occur with certain aspectual markers from Table 2. This way, verbs are automatically ranked according to their "degree of stativity."

Machine learning has become instrumental in the development of robust natural language understanding systems in general (Cardie and Mooney 1999). For example, decision tree induction has been applied to word sense disambiguation (Black 1988), determiner prediction (Knight et al. 1995), coordination parsing (Resnik 1993), syntactic parsing (Magerman 1993), and disambiguating clue phrases (Siegel 1994; Siegel and McKeown 1994; Litman 1994). An overview of psycholinguistic issues behind learning for natural language problems in particular is given by Powers and Turk (1989). Models resulting from machine induction have beeen manually inspected to discover linguistic insights for disambiguating clue words (Siegel and McKeown 1994). However, machine learning techniques have not previously been applied to aspectual disambiguation.

Previous efforts have applied machine induction methods to coordinate corpus-based linguistic indicators in particular, for example, to classify adjectives according to markedness (Hatzivassiloglou and McKeown 1995), to perform accent restoration (Yarowsky 1994), for sense disambiguation problems (Luk 1995), and for the automatic identification of semantically related groups of words (Pereira, Tishby, and Lee 1993; Hatzivassiloglou and McKeown 1993; Schütze 1992).

8. Future Work

Parallel bilingual corpora are potential sources of supervised examples for training and testing aspectual classification systems. For example, since many languages have explicit markings corresponding to completedness (as described in Section 2.6), the category of a clause can be determined by its translation.

Additional machine learning methods should be evaluated for combining linguistic indicators. For example, neural networks are especially suited for combining numerical inputs, and Naive Bayes models are especially suited for additive concepts. Also, iteratively refining the model (e.g., for logistic regression) may be an important way to eliminate indicators that do not help for a particular classification problem and to eliminate redundancy between indicators that correlate highly with one another.

Machine learning techniques may be able to automatically determine how best to measure linguistic indicators, if trained over a large supervised sample. For example, previous work has measured indicators by applying a symbolic expression induced by GP to a subset of clauses in a corpus (Siegel and McKeown 1996). This way, interactions between markers in a clause can be automatically measured. In principle, machine learning techniques could further generalize these methods by automatically inducing

an algorithm that scans a corpus dynamically, depending on what it sees as it processes clauses. This could automatically select relevant markers as well as relevant portions of the corpus for a particular input clause.

9. Conclusions

While individual linguistic indicators have predictive value, they are predictively incomplete. Such incompleteness is due to sparsity and noise when computing indicator values over a corpus of limited size, and is also a consequence of the linguistic behavior of certain indicators. However, incomplete indicators can complement one another when placed in combination.

Machine learning has served to illustrate the potential of 14 linguistic indicators by showing that they perform well in combination for two aspectual classification problems. This potential was not clear when evaluating indicators individually. For stativity, decision tree induction achieved an accuracy of 93.9%, as compared to the uninformed baseline's accuracy of 83.8%. Furthermore, GP and logistic regression also achieved improvements over the baseline. For completedness, CART and logistic regression achieved accuracies of 74.0% and 70.5%, as compared to the uninformed baseline's accuracy of 63.3%. These improvements in classification performance are more dramatically illustrated by favorable trade-offs between recall scores achieved for both classification problems. Such results are profitable for tasks that weigh the identification of the less frequent class more heavily.

This evaluation was performed over unrestricted sets of verbs occurring across two corpora. The system can automatically classify all verbs appearing in a corpus, including those that have not been manually classified for supervised training data. Therefore, we have demonstrated a much-needed full-scale method for aspectual classification that is readily expandable. Since minimal overfitting was demonstrated with only a small quantity of manually supervised data required, this approach is easily portable to other domains, languages, and semantic distinctions.

The results of learning are linguistically viable in two respects. First, learning automatically produces models that are specialized for different aspectual distinctions; the same set of 14 indicators are combined in different ways according to which classification problem is targeted. Second, automatic learning often derives linguistically informative insights. We have shown several such insights revealed by inspecting the models produced by learning, which are summarized here:

- Examining the logistic regression model for classification according to stativity revealed a decision tree type of rule incorporated with the normal weighting scheme.
- Verb frequency distinguishes stative verbs within multiple subsets of verbs. When applied to all verbs in a medical corpus, it identifies occurrences of *show*. Furthermore, examining an example node of the decision tree that distinguishes according to stativity revealed that verb frequency discriminates 19 stative clauses with 100.0% precision from the node's partition of 60 training cases.
- Several proper subsets of the linguistic indicators prove independently useful for aspectual classification when combined with an appropriate model. This is illustrated by the fact that certain models reveal combinations of small sets of indicators that improved classification performance. For example, GP results for both classification tasks

incorporated a subset of only five indicators each. In particular, manner adverb, which ranked highest by logistic regression, is not incorporated in the example function tree induced by GP. This may be because this indicator only applies to a small number of verbs, as shown in Table 13, and because an *if*-rule such as that captured by logistic regression is difficult to encode with a function tree with no conditional primitives.

- Learning methods discovered that some indicators are particularly useful for both classification tasks. For example, the same two indicators were weighted most heavily by logistic regression for both tasks: duration *in*-PP and manner adverb.
- However, in general, learning methods emphasized different linguistic indicators for different classification tasks. For example, decision tree induction used `frequency` as the main discriminator to classify clauses according to stativity, while the perfect indicator was the main discriminator for classification according to completedness.

Comparing the ability of learning methods to combine linguistic indicators is difficult, since they rank differently depending on the classification task and evaluation criteria. For example, the relative accuracies of the three supervised learning procedures rank in opposite orders when comparing the results for stativity to the results for completedness.

The unsupervised grouping of verbs provides an additional method for aspectual classification according to stativity. Co-occurrence distributions between the verb and direct object inform the aspectual classification of verbs. This provides information beyond the 14 linguistic indicators that can also be derived automatically. However, due to the sparsity intrinsic to pairs of open-class categories such as verb-object pairs, this approach was only evaluated over a small set of frequent verbs.

Acknowledgments

Judith Klavans was very helpful in our formulation of the linguistic techniques in this work, and Alexander Chaffee, Vasileios Hatzivassiloglou, and Dekai Wu in the results evaluation methods. Vasileios Hatzivassiloglou designed and implemented the clustering of verbs described in Section 6. For comments on earlier drafts of this work, we would like to thank those people mentioned, as well as David Evans, Hongyan Jing, Min-Yen Kan, Dragomir Radev, and Barry Schiffman. Finally, we would like to thank Andy Singleton for the use of his GPQuick software.

This research was supported in part by the Columbia University Center for Advanced Technology in High Performance Computing and Communications in Healthcare (funded by the New York State Science and Technology Foundation), the Office of Naval Research under contract N00014-95-1-0745, and the National Science Foundation under contract GER-90-24069.

References

Allen, Franklin and Risto Karjalainen. 1995. Using genetic algorithms to find technical trading rules. Technical Report, Rodney L. White Center For Financial Research.

Allen, James. 1995. *Natural Language Understanding*. Benjamin/Cummings, Redwood City, CA.

Baker, R. J. and J. A. Nelder. 1989. *The GLIM System, Release 3: Generalized Linear Interactive Modeling*. Numerical Algorithms Group, Oxford.

Black, Ezra. 1988. An experiment in computational discrimination of English word senses. *IBM Journal of Research and Development*, 2(32).

Breiman, Leo, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone. 1984. *Classification and Regression Trees*. Wadsworth, Belmont.

Brodley, Carla E. 1996. Applying classification algorithms in practice. In *Statistics and Computing*.

Cardie, Claire and Nicholas Howe. 1997. Improving minority class prediction using

case-specific feature weights. In D. Fisher, editor, *Proceedings of the Fourteenth International Conference on Machine Learning*. Morgan Kaufmann.
Cardie, Claire and Raymond J. Mooney. 1999. Guest editors' introduction: Machine learning and natural language. *Machine Learning*, 1–3(34).
Chan, Philip K. and Sal J. Stolfo. 1993. Toward multistrategy parallel and distributed learning in sequence analysis. In *Proceedings of the First International Conference on Intelligent Systems for Molecular Biology*.
Church, Ken. 1988. A stochastic parts program and noun phrase parser for unrestricted text. In *Proceedings of the 2nd Conference for Applied Natural Language Processing*, pages 136–143.
Cohen, William W. 1995. Fast effective rule induction. In *Proceedings of the 12th International Conference on Machine Learning*, pages 115–123.
Cramer, Nichael L. 1985. A representation for the adaptive generation of simple sequential programs. In J. Grefenstette, editor, *Proceedings of the [First] International Conference on Genetic Algorithms*. Lawrence Erlbaum.
Dorr, Bonnie J. 1992. A two-level knowledge representation for machine translation: lexical semantics and tense/aspect. In James Pustejovsky and Sabine Bergler, editors, *Lexical Semantics and Knowledge Representation*. Springer Verlag, Berlin.
Dowty, David R. 1979. *Word Meaning and Montague Grammar*. D. Reidel, Dordrecht.
Duda, Richard O. and Peter E. Hart. 1973. *Pattern Classification and Scene Analysis*. Wiley, New York.
Fayyad, Usama M. and Keki B. Irani. 1992. On the handling of continuous-valued attributes in decision tree generation. *Machine Learning*, 8.
Friedman, Jerome H. 1977. A recursive partitioning decision rule for non-parametric classification. *IEEE Transactions on Computers*.
Goldberg, David. 1989. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Publishing Company, Inc., Reading, MA.
Hatzivassiloglou, Vasileios. 1997. *Automatic Acquisition of Lexical Semantic Knowledge from Large Corpora: The Identification of Semantically Related Words, Markedness, Polarity, and Antonymy*. Ph.D. thesis, Columbia University.
Hatzivassiloglou, Vasileios and Kathleen R. McKeown. 1993. Towards the automatic identification of adjectival scales: clustering adjectives according to meaning. In *Proceedings of the 31st Annual Meeting*, pages 172–182, Columbus, OH, June. Association for Computational Linguistics.
Hatzivassiloglou, Vasileios and Kathleen R. McKeown. 1995. A quantitative evaluation of linguistic tests for the automatic prediction of semantic markedness. In *Proceedings of the 33rd Annual Meeting*, pages 197–204, Boston, MA, June. Association for Computational Linguistics.
Hitzeman, Janet, Marc Moens, and Claire Grover. 1994. Algorithms for analysing the temporal structure of discourse. Technical Report, University of Edinburgh.
Holland, John. 1975. *Adaptation in Natural and Artificial Systems*. The University of Michigan Press, Ann Arbor, MI.
Hwang, Chung Hee and Lenhart K. Schubert. 1991. Interpreting temporal adverbials. Technical Report, University of Rochester.
Klavans, Judith L. 1994. Linguistic tests over large corpora: aspectual classes in the lexicon. Technical Report, Columbia University Dept. of Computer Science.
Klavans, Judith L. and Martin Chodorow. 1992. Degrees of stativity: the lexical representation of verb aspect. In *Proceedings of the 14th International Conference on Computation Linguistics*.
Knight, K., I. Chander, M. Haines, V. Hatzivassiloglou, E. Hovy, M. Iida, S. K. Luk, R. Whitney, and K. Yamada. 1995. Filling knowledge gaps in a broad-coverage mt system. In *Proceedings of the International Joint Conference on Artificial Intelligence*.
Koza, John R. 1992. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA.
Litman, Diane J. 1994. Classifying cue phrases in text and speech using machine learning. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, Menlo Park, CA, July. AAAI Press.
Luk, Alpha K. 1995. Statistical sense disambiguation with relatively small corpora using dictionary definitions. In *Proceedings of the 33rd Annual Meeting*, Columbus, OH, June. Association for Computational Linguistics.
Magerman, David H. 1993. Parsing as statistical pattern recognition. Technical Report, IBM.
Masand, Brij. 1994. Optimizing confidence of text classification by evolution of

symbolic expressions. In K. Kinnear, editor, *Advances in Genetic Programming*. MIT Press, Cambridge, MA.
McCord, Michael C. 1990. SLOT GRAMMAR: A system for simpler construction of practical natural language grammars. In R. Studer, editor, *International Symposium on Natural Language and Logic*. Springer Verlag.
Mitchell, Tom M. 1997. *Machine Learning*. The McGraw-Hill Companies, Inc., New York.
Moens, Marc and Mark Steedman. 1988. Temporal ontology and temporal reference. *Computational Linguistics*, 14(2).
Olsen, Mari B. and Philip Resnik. 1997. Implicit object constructions and the (in)transitivity continuum. In *Proceedings of the 33rd Regional Meeting of the Chicago Linguistics Society*, April.
Passonneau, Rebecca J. 1988. A computational model of the semantics of tense and aspect. *Computational Linguistics*, 14(2).
Pereira, Fernando, Naftali Tishby, and Lillian Lee. 1993. Distributional clustering of English words. In *Proceedings of the 31st Annual Meeting*, pages 183–190, Columbus, OH, June. Association for Computational Linguistics.
Powers, David M. W. and Christopher C. R. Turk. 1989. *Machine Learning of Natural Language*. Springer-Verlag.
Pustejovsky, James. 1991. The syntax of event structure. *Cognition*, 41(103):47–92.
Pustejovsky, James. 1995. *The Generative Lexicon*. MIT Press, Cambridge, MA.
Quinlan, Jim R. 1986. Induction of decision trees. *Machine Learning*, 1(1):81–106.
Quinlan, Jim R. 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA.
Resnik, Philip. 1993. Semantic classes and syntactic ambiguity. In *Proceedings of the ARPA Workshop on Human Language Technology*, March.
Resnik, Philip. 1996. Selectional constraints: An information-theoretic model and its computational realization. *Cognition*, (61).
Schütze, Hinrich. 1992. Dimensions of meaning. In *Proceedings of Supercomputing*.
Schubert, Lenhart K. and Chung Hee Hwang. 1990. Picking reference events from tense trees: A formal, implementable theory of English tense-aspect semantics. Technical Report, University of Rochester.
Siegel, Eric V. 1994. Competitively evolving decision trees against fixed training cases for natural language processing. In K. Kinnear, editor, *Advances in Genetic Programming*. MIT Press, Cambridge, MA.
Siegel, Eric V. 1997. Learning methods for combining linguistic indicators to classify verbs. In *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*, Providence, RI, August.
Siegel, Eric V. 1998a. Disambiguating verbs with the WordNet category of the direct object. In *Proceedings of the Usage of WordNet in Natural Language Processing Systems Workshop*, Montreal, Canada.
Siegel, Eric V. 1998b. *Linguistic Indicators for Language Understanding: Using Machine Learning Methods to Combine Corpus-based Indicators for Aspectual Classification of Clauses*. Ph.D. thesis, Columbia University.
Siegel, Eric V. 1999. Corpus-based linguistic indicators for aspectual classification. In *Proceedings of the 37th Annual Meeting*. Association for Computational Linguistics.
Siegel, Eric V. and Kathleen R. McKeown. 1994. Emergent linguistic rules from inducing decision trees: disambiguating discourse clue words. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, Menlo Park, CA, July. AAAI Press.
Siegel, Eric V. and Kathleen R. McKeown. 1996. Gathering statistics to aspectually classify sentences with a genetic algorithm. In K. Oflazer and H. Somers, editors, *Proceedings of the Second International Conference on New Methods in Language Processing*, Ankara, Turkey, September, Bilkent University.
Sjantner, T. J. and D. E. Duffy. 1989. *The Statistical Analysis of Discrete Data*. Springer-Verlag, New York.
Stolfo, Salvatore, Andreas L. Prodromidis, Shelley Tselepis, Wenke Lee, and Wei Fan. 1997. JAM: Java agents for meta-learning over distributed databases. Technical Report, Columbia University.
Stys, Malgorzata E. 1991. Parallel science texts in English and Polish: Problems of language and communication. Technical Report, Warsaw University. Master's thesis.
Syswerda, Gibert. 1989. Uniform crossover in genetic algorithms. In J. D. Schaffer, editor, *Proceedings of the Third International Conference on Genetic Algorithms*. Morgan Kaufmann.
Tackett, Walter A. 1993. Genetic programming for feature discovery and image discrimination. In *Proceedings of the Fifth International Conference on Genetic Algorithms*, San Mateo, CA. Morgan Kaufmann.
Tackett, Walter A. and Aviram Carmi. 1994. The donut problem: Scalability,

generalization and breeding policies in genetic programming. In K. Kinnear, editor, *Advances in Genetic Programming*. MIT Press, Cambridge, MA.
Vendler, Zeno. 1967. Verbs and times. In *Linguistics in Philosophy*. Cornell University Press, Ithaca, NY.
Wiebe, Janyce M., Thomas P. O'Hara, Thorsten Öhrström-Sandgren, and Kenneth J. McKeever. 1997. An empirical approach to temporal reference resolution. In *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*, Providence, RI, August.
Wolpert, David H. 1992. Stacked generalization. *Neural Networks*, 5.
Yarowsky, David. 1994. Decision lists for lexical ambiguity resolution: Application to accent restoration in Spanish and French. In *Proceedings of the 32nd Annual Meeting*, San Francisco, CA, June. Morgan Kaufmann. Association for Computational Linguistics.