Lexical Aspectual Classification

by

Richard Keelan

Thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
In partial fulfillment of the requirements
For the degree of Master of Computer Science (MCS)

Ottawa-Carleton Institute for Computer Science
School of Electrical Engineering and Computer Science
University of Ottawa

Abstract

This work is a first attempt at classification of Lexical Aspect. In this dissertation I describe eight lexical aspectual classes, each initially containing a few members. Using distributional analysis I generate 132 additional seeds, each of which was approved by at least seven out of nine judges. These seeds are in turn fed into a supervised machine learning system, trained on 136 lexical and syntactic features. I experiment on one 8-way classification task, one 3-way classification task, and ten binary classification tasks, and show that five of the eight classes are identified better than by a random baseline measure by a statistically significant margin. Finally, I analyze the relative contribution of each of four feature groups and conclude that the same features which are best in identifying phrasal aspect are also most informative for lexical aspect.

Acknowledgements

Numerous people aided and abetted the writing of this dissertation, and I would like to devote a few lines to thanking them.

I thank Dr. Stan Szpakowicz for supervising me closely enough that I succeeded but not so closely that I did not have to think for myself; and Dr. Marina Sokolova for learned advice, especially in the realm of Machine Learning.

I would like to thank my colleagues from the School of Electrical Engineering and Computer Science, Alistair, Anna, Martin, and Chris—for advice and insightful discussion. I must also thank my girlfriend, Natasha, for supporting me throughout this process, especially during the final weeks leading up to submission.

Finally, I would like to thank all the people who judged my automatically acquired seeds: Justin Girard, Val Keelan, Alistair Kennedy, Mark Binette, Dawn and John Merriam, Bryan Keelan, Geoff Keelan, and Alayna Schulle.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Automatic verb classification as a research area

Shallow approaches to NLP, sometimes known as 'word counting', treat words as mere strings of characters, ignoring any information embedded in the semantics of words, or in the grammatical rules that govern sentence structure. Any deeper analysis of language will require knowing properties of words: their meaning, how they may be combined to form larger structures such as clauses and sentences, and how such combinations affect meaning. Lexical acquisition is the process of learning the properties of words. Lexical acquisition can be done manually, as in the case of *WordNet* (Fellbaum, 1998), or automatically. The promise of automatic lexical acquisition is that lexical resources can be built quickly and cheaply, and thereby allow researchers to create lexical resources for other languages that match the scale and breadth of those available for English, or allow researchers to produce lexicons tailored for a specific domain or problem, rather than use a general resource for a domain specific problem.

Within lexical acquisition, learning the properties of verbs is particularly important because verbs convey how other constituents of a sentence—such as subjects, objects, adverbs, and prepositional phrases—relate to one another. One method for learning the

properties of verbs is classification, grouping together verbs because they share semantics, or aspectual structure, or syntax. *Aspectual classifications*, such as Vendler's (1967), group verbs together based on sharing the same aspectual viewpoint and internal event structure. *Lexical semantic classifications*, such as Levin's (1993), group verbs together based on sharing a combination of lexical and semantic properties. In this dissertation, I consider a set of classes proposed by Leech (2004),[1] which are based on a combination of aspect and meaning.

Work in the field of automatic verb classification varies primarily on the method of classification, and the criteria upon which the classification is based. The methods of classification can be summed up as rule-based (Passonneau, 1988; Brent, 1991; Dorr, 1997b; Korhonen, 2002), unsupervised learning (Schulte Im Walde, 2000; Kingsbury and Kipper, 2003; Green and Dorr, 2004; Korhonen et al., 2006; Schulte Im Walde, 2006; Schulte Im Walde et al., 2008; Vlachos et al., 2009; Sun, 2011), or supervised learning (Siegel and McKeown, 2000; Joanis et al., 2006; Li and Brew, 2008; Brown et al., 2011). Rule-based methods dominated at first, but gave way to machine learning based methods, of which unsupervised methods are more popular.

The other broad discriminator of automatic verb classification is the target classification, or the criteria upon which verbs are being classified. Levin's (1993) English Verb Classes and Alternations is by far the most popular target classification (Dorr and Jones, 1996; Dorr, 1997b; Schulte Im Walde, 2000; Korhonen, 2002; Joanis et al., 2006; Li and Brew, 2008; Vlachos et al., 2009; Sun, 2011). Classifying according to lexical resources such as *FrameNet* (Green and Dorr, 2004; Schulte Im Walde, 2006) and *VerbNet* (Kingsbury and Kipper, 2003; Brown et al., 2011) is also popular. Finally, there have been some attempts to classify according to aspect (Passonneau, 1988; Brent, 1991; Siegel and McKeown, 2000).

[1] Slightly modified, as described below.

1.2 Leech's classes

Leech proposes the following classes: momentary verbs, transitional event verbs, activity verbs, process verbs, inert perception verbs, inert cognition verbs, attitude verbs, having and being verbs, and bodily sensation verbs. Each class has a handful example verbs and a very brief description.

In order to use Leech's classes as the foundation for an automatic verb classification system, I made a few slight modifications (see **Table 1.1**). I changed the name of the process verbs to *Change* because they are more accurately characterized by the notion of *change* than of *process*. Building a house is also a process, but that verb is properly characterized as an *Activity*. In renaming the having and being verbs I expanded the scope of the class to include all verbs denoting a stable relationship. I also expanded the scope *Cognition* and *Perception* verbs to include non-inert verbs, which Leech characterizes as *Activities*. The rationale for this decision is that the similarity between, for example, active and inert perception verbs is greater than that between active perception verbs and other *Activities*. Furthermore, I believed that the similarity between active and inert perception verbs would be more interesting to other researchers wishing to use this classification.[2] I also merged the verbs of *perception* and *bodily sensation* on the rationale that bodily sensation is a type of perception. A final modification is that I use this as a classification of verb types.

Of other classifications that have been attempted, Leech's is most similar to aspectual classification, since it has a similar number of classes and is similarly broad. Like an aspectual classification, it ideally covers all verbs; whereas Levin's EVCA only covers those verbs whose diathesis alternations had been extensively investigated (Levin, 1993, p. 18). Leech's classes have been used in previous research in negotiation classification (Sokolova and Szpakowicz, 2006) and opinion analysis (Sokolova and Lapalme, 2008). I discuss Leech's classes in greater detail in the following chapter.

[2]The same rationale holds for *Cognition* verbs.

Modified Class	**Original Class**
Transitional Events	Transitional Event verbs
Momentary Events	Momentary verbs
Activity	Activity verbs
Change	Process verbs
Perception	Inert Perception verbs, Bodily Sensation verbs
Cognition	Inert cognition verbs
Attitude	attitude verbs
Relationship	having and being verbs

Table 1.1: Modified Classes

1.3 Motivation

Why would one bother classifying verbs? One way of approaching NLP is to know about words. If a system knew the precise meaning of every word, and precisely the rules governing its use, then it would certainly be able to understand language, and communicate. Knowing everything about every word, though, is hard; and full understanding of natural language is perhaps an overly ambitious goal. Classifying words allows one to know some things about some words, which is a much more manageable problem.

For example, dividing the set of all words into 36 groups consisting of varieties of nouns, verbs, descriptors (Adjectives and Adverbs), and function words is useful, and not impossibly difficult. Modern part of speech (POS) taggers can achieve better than 97% accuracy (Toutanova et al., 2003), and POS tagging is a first step in many algorithms for parsing. POS tagging is both (relatively) easy and useful because of abstraction. Broadly speaking, abstraction means hiding many irrelevant details in order to focus on a few highly relevant details. POS tagging is easy, then, because a person or system need only answer one question about each word: What grammatical category does it belong to? Furthermore, abstraction makes POS tagging useful because knowing the POS tag of a word tells a person or system one thing unambiguously; which is much preferred to knowing many ambiguous things about a word. Some parsing algorithms, for example,

require on input the POS tags of each word in the sentence.

Word classification is useful by the same principle. As Korhonen (2010) notes in a recent review of automatic lexical acquisition, word classes can alleviate the sparse data problem by predicting that a classified-but-rare word will behave similarly to a classified-but-frequent word. Verbs are particularly interesting because they tie the other components of the sentence together—they "define the mapping from surface realization of arguments to predicate-argument structure" (Kipper et al., 2008). Verb classification can be a tool in many other NLP problems, such as word sense disambiguation, semantic role labeling, information extraction, question answering, machine translation, and summarization (Kipper et al., 2008).

1.4 Making a verb classification operational

Leech's classification represents an interesting problem. Each class is described very sparingly—a suggestive name, a couple words of elaboration,[3] and a handful of examples. Yet, based on this alone, native speakers of English will, when presented with a new verb and a class, be able to judge whether the verb belongs in that class. *Ache*, *feel*, *hear*, *hurt*, *itch*, *see*, *smell*, *taste*, *tingle*—one knows what each verb means, just as one knows that the element common to all of them is the use of the senses. One can further conclude that *feel* in the sense of 'believe' (*e.g.*, I feel that this is the incorrect course of action) does not belong in this class, because it does not share that common element.

Making a verb classification operational requires uncovering all the implicit knowledge a native speaker of English uses to judge which class a verb belongs in. It requires first defining the boundaries of the classes in question, and then assigning members to each class. The difficulty is in determining which properties of words lie at the boundaries between classes—which properties can serve to distinguish between them. Furthermore, since the distinguishing properties may be deep semantic properties of words or "rela-

[3]For example, "*[Momentary events] refer to happenings so momentary that it is difficult to think of them as having duration*" or "*[Activity verbs] tell us something is 'going on'*".

tions" between words, it may be necessary to come up with reliable methods for indirectly detecting their presence or absence. Leech has proposed a classification of verbs by aspect and semantics. This dissertation aims to prove the validity of his classes by uncovering and making operational the knowledge required to assign verbs to them automatically.

Making Leech's classes operational is made all the more difficult by the fact that many of the example verbs are highly polysemous. Native speakers of English are able to determine the shared semantics of each class while filtering out irrelevant senses of each word, but state of the art NLP is not. For example, the verb *feel* can refer to a physical sensation (belonging therefore in the Perception class) or a belief (belonging therefore in the Cognition class). By contrast, the verb *play* can refer to the action of using an instrument (play the drums), or participating in a game (play soccer). Native speakers of English can tell that only the first sense of *feel* belongs in the Perception class, while both sense of *play* belong in the Activity class.

1.5 Structure of the dissertation

The rest of this dissertation proceeds in the following manner. **Chapter 2** discusses the linguistic background. **Chapter 3** surveys related literature. **Chapter 4** describes my work in semi-automatically expanding my seed set. **Chapter 5** describes describes my use of Machine Learning for automatic verb classification. Finally, **Chapter 6** concludes by stating the contributions of this dissertation and outlining some future work.

Chapter 2

Linguistic Background

2.1 Chapter Overview

This chapter presents an overview of linguistics research relating to verb classification. In the following sections I will discuss six linguistically motivated verb classifications, including Levin's (1993) English verb classes and alternations (**Section 2.2**), four aspectual classifications (**Section 2.3**), and Leech's (2004) verb classes (**Section 2.4**).

I focus more on the aspectual classifications, because they bear greater similarity to Leech's, upon which this dissertation is based. Levin's work produces classifications which are very fine-grained, because it accounts for many properties of verbs. The aspectual classifications, by contrast, are very coarse, because they focus primarily on one specific facet of meaning, in this case temporal aspect.

A coarse classification which isolates the one or two properties one is interested in can be easier to use than a fine-grained classification, because a fine-grained classification requires one to first disregard distinctions which are irrelevant to the task at hand. I discuss fine-grained classifications as well because they have received more attention in the automatic verb classification literature, because they can offer insight into the

task of coarse-grained classification, and because they serve directly as a resource for coarse-grained classifications. For example, Dorr and Olsen (1997) use the English Verb Classes and Alternations (EVCA) (fine-grained) to determine a verb's Lexical Conceptual Structure (LCS), and the LCS to determine its aspectual class (coarse-grained).

2.2 English Verb Classes and Alternations

Levin's (1993) English Verb Classes and Alternations (EVCA) formulates classes of verbs by examining which diathesis alternations the verbs allow. Diathesis alternations are differences in how a verb predicate maps syntactic arguments (subject, object, etc.) to semantic arguments (agent, instrument, etc). This approach is different from the one taken by *WordNet* because it relies on syntactic as well as semantic information, whereas *WordNet* relies exclusively on semantic infomration.

The motivation behind EVCA is to identify meaning components that contribute to the overall meaning of a group of verbs (Levin, 1993, p. 18). How meaning components can be identified based on diathesis alternations is best explained by an example.

Levin (1993, p. 5) examines four transitive verbs, *touch*, *hit*, *cut*, and *break*, noting that only *break* does not allow the *body-part possessor ascension alternation*. Levin further notes that among the four verbs only *break* does not necessarily imply contact as part of the meaning of the verb. Levin then concludes that the other three verbs allow the *body-part possessor ascension alternation* because they have the notion of *contact* as part of their meaning.

The verb classes are defined by diathesis alternations, which are divided into 7 groups. The first group, *transitivity alternations* discuss two alternations, those in which the object of the verb is dropped ('NP VP NP' vs 'NP V'), and those in which a noun phrase is replaced with a prepositional phrase ('NP VP NP' vs 'NP V PP'). These alternations are subdivided into *subject and object alternations*, *unexpressed object alternations*, *conative alternations*, and *preposition drop alternations*. One specific alternation in this group is

the Causative/Inchoative alternation, which generally characterizes change of state or location verbs. To illustrate, "Janet broke the cup" could also be expressed as "The cup broke". This is one of the *subject-object alternations*, with "the cup" serving as the object in the first case, and subject in the second (Levin, 1993, p. 29).

The *unspecified object alternation* is one of the *unexpressed object alternations*. An illustrative example from (Levin, 1993, p. 33) is:

(a) Mike ate the cake.

(b) Mike ate.

The alternation manifests with activity verbs, and tends to have a typical object. For "eat", the typical object is an edible item. Even when the object is dropped, it is implied that Mike is eating something edible.

The second alternation group comprises 14 subgroups involving arguments within the verb phrase, but without a change of transitivity. Instead it is defined by alternations displayed by transitive verbs with more than one argument in the verb phrase. These alternations allows more than one expression of the internal arguments.[1]

One subgroup is the *possessor-attribute factoring alternation*, in which a possessor and a possessed attribute are expressed in different ways with a verb. The *possessor object alternation*, a member of the aforementioned group, works with many psychological verbs, but disallows verbs of perception, like "see" or "hear".

(a) I admired his courage.

(b) I admired him for his courage.

As the example shows, the alternation hinges on the object of the verb being expressed as a single noun phrase in (a), and as a pair of noun phrases, expressing the possessor and its attribute in (b).

[1] internal to the verb phrase

This alternation is very similar to another in this subgroup, the *body-part possessor ascension alternation*. The differences are that they allow different types of verbs and, happily, that the first uses the preposition *for*, while the second uses a locative preposition.

"Oblique" subject alternations involve verbs with agentive subjects. In one case, the verb takes the agent as its subject, and a noun phrase expressed in a prepositional phrase as the object. In the other case, the noun from the prepositional phrase becomes the subject, and the agent is dropped. Consider "He established his innocence with the letter", for example. "The letter" is the object, expressed in a prepositional phrase headed by *with*, and becomes the subject in this alternate form "The letter established his innocence". The 10 alternations in this group are differentiated by the 'class' of the "oblique" subject.[2] In this case the "oblique" subject is the letter, which is an abstract cause (Levin, 1993, p. 81).

Reflexive diathesis alternations involve replacing the subject with the object, and the object with a reflexive pronoun. For example, "The butler polishes the silver" alternates with "This silver polishes itself" (Levin, 1993, p. 89).

Four alternations use the passive voice: *verbal passive*, *prepositional passive*, *adjectival passive*, and *adjectival perfect participles*. An example of a *verbal passive alternation* is "The police kept tabs on the suspect", compared to "Tabs were kept on the suspect".

Alternations involving postverbal subjects are constructions where the subject of the verb appears after the verb, built either from 'there' insertion or locative inversion. 'There' insertion most often turns an intransitive verb into a transitive verb with 'there' as a dummy subject. For example, "A problem developed" becomes "There developed a problem" (Levin, 1993, p. 89). Locative inversion, on the other hand, places a prepositional phrase in front of the verb instead of the subject. Many verbs of existence demonstrate this alternation, for example, "On the windowsill is a flowering plant" could also, and perhaps more naturally, be put "A flowering plant is on the windowsill" (Levin,

[2]Class is a mixed bag of theta-roles (instrument, abstract cause, location) and entities of varying abstractness (time, natural force, raw material).

1993, p. 92).

The seventh group is labeled "Other Constructions", and includes constructions involving the selectional preferences of verbs, such as the ability for certain verbs to take cognate objects ("Sarah sang a song") (Levin, 1993, p. 95) There is also an eighth section on verbs with special diathesis restrictions, such as when verbs require a reflexive pronoun for the object ("The politician perjured himself")(Levin, 1993, p. 107).

These alternations produce more than 150 classes split into 49 groups, including verbs of creation and transformation, verbs of perception, verbs of psychological state, verbs of desire, verbs of communication, verbs involving the body, verbs of change of state, verbs of existence, and aspectual verbs. Each of these classes is characterized by specific alternations and other argument-taking properties.

For example, verbs of psychological state are split in 4 subgroups: *amuse* verbs, *admire* verbs, *marvel* verbs, and *appeal* verbs. To take one example, *amuse* verbs are transitive and have the experiencer as the **object**, while the *admire* verbs are transitive and have the experiencer as the **subject**. *Amuse* verbs allow the middle alternation, but not causative alternations (Levin, 1993, p. 190).

2.3 Aspectual Classifications

The aspect of a verb refers to the internal event structure (Aktionsart) of a verb, as well as its presentation (e.g., either perfective or imperfective) (Smith, 1991, p. 3). Aristotle was the first to make an aspectual distinction among verbs, distinguishing *kineseis* ("movements") and *energia* ("actualities"), which roughly corresponds to the telic-atelic distinction.[3] However, most aspectual classifications make a first cut at the difference between states and non-states (also called events). The topic was later discussed by Ryle (1949), Kenny (1963), Vendler (1967) and Dowty (1979).

[3]Quoted after Dowty (1979).

Context	**Name**
State-E-Same state	Happening
State-E-Different state	Transition
Process-E-State	Culmination
Process-E-Same process	Disturbance
State-E-Process	Activation
Process-E-Different process	Switch

Table 2.1: Aspectual Histories—Events

2.3.1 Aspectual Classes of Histories

Nakhimovsky (1988) describes two separate classifications of verbs, one for events and one for non-instantaneous histories (states and processes). These aspectual classes are properties of *histories*, which are situations evolving or persisting over time. Histories can be of the *type* or *token* variety. *History-types* are generic references, while *history-tokens* are specific instances of *history-types* (Nakhimovsky, 1988, p. 30). The aspectual classes of a *history-type* and *history-token* roughly correspond to the difference between lexical aspectual class and phrasal aspectual class, which I discuss further in **Section 2.3.2**.

Events are classified according to what precedes and follows them (see **Table 2.1**), while non-instantaneous histories are classified according to 3 criteria:

1. Internal dynamics.
2. Telicity.
3. Resources consumed.

This leads to the following five classes:

1. Zero-resource states (e.g., knowing English, owning a house).

2. Generic-resource states (e.g., sleeping).
3. Atelic processes consuming generic resources.
4. Atelic processes consuming generic and specific resources.
5. Telic processes.

Telicity applies to processes but not to states. A general resource is a property of the entity involved in a history, while a process-specific resource is a property of the process itself. A human walking, for instance, will only walk so long as he or she has energy to keep moving. The energy which is consumed is a property of the person. By contrast, reading a book can only occur so long as the reader is awake (a generic resource), and there is material to read (a process-specific resource).

Nakhimovsky refers to two distinctions among processes: the consumption or lack of consumption of process-specific resources, and the telicity of a process. Telic processes are "*processes that have a built-in terminal point that is reached in the normal course of events and beyond which the process cannot continue*"(Nakhimovsky, 1988, p. 34). Furthermore, Nakhimovsky states that most telic processes are either human activities directed towards a goal, or processes that consume a specific amount of a resource specific to that process (and further that these categories overlap) (1988, p. 34).

Presumably, the difference between telic and atelic processes is the explicitness of the terminal point, not the existence of it, since an atelic process which consumes specific resources (*reading*, to use one of Nakhimovsky's examples) will end once that resource runs out. Indeed, *read*, on its own, is classified by Nakhimovsky as an atelic, specific-resource verb, while "*read a book*" is classified as a telic verb (1988, p. 36). Similarly, the other atelic, specific-resource verb examples, *write*, *build* and *dig*, all become telic verbs when combined with an object (letter, chair, and hole, respectively).

Some examples of atelic processes consuming generic resources are *walk*, *run*, and *work*. Some examples of generic-resource consuming states are *sleep*, *stand*, *sit*, *lie*, and *hold*.

Nakhimovsky splits zero-resource states into three groups:

1. Relations: own, possess, resemble.
2. Perceptions: see, hear, feel.
3. Mental states: know, remember trust.

Nakhimovsky points out that Vendler's (1967) classification is based on language cues such as whether a verb takes the progressive form, and that Dowty's (1979) classification depends on the truth of a sentence at an interval and its subintervals. Nakhimovsky, meanwhile, argues that a classification of language should depend on something perceived or experienced, not on the truth value of a sentence (Nakhimovsky, 1988, p. 34).

This classification is proposed in order to better understand narratives, so knowing that a state consumes generic resources, and will therefore only hold while that resource is present, will allow a very deep understanding of a narrative. Unfortunately, it also relies on deep semantic knowledge, such as (for example) knowing that sleep is a resource-consuming state, whereas ownership of a house is not.

2.3.2 Temporal Ontology

Moens and Steedman (1988) propose a classification of English propositions into **aspectual types** which they define (somewhat awkwardly) as "the relation that a speaker predicates of the particular happening that their utterance describes, relative to other happenings in the domain of discourse." In other words, aspectual type describes how an event is related to other co-occurring events; and more specifically it describes the speaker's portrayal of the event rather than the underlying reality.[4]

Table 2.2 shows the classes of the classification, the dimensions which yield them, and some examples.[5] In this account, the distinction between *events* and *states* is that

[4]In this case, as in others, writers are understood to be subsumed under the concept of speakers.

[5]Examples taken from (Moens and Steedman, 1988, p. 17).

Class Name	**Consequential**	**Atomic**	**Stative**	**Examples**
Culmination	Yes	Yes	No	recognize, spot, win the race
Culminated Process	Yes	No	No	build a house, eat a sandwich
Point	No	Yes	No	hiccup, tap, wink
Process	No	No	No	run, swim, walk, play the piano
State	N/A	N/A	Yes	understand, love, know, resemble

Table 2.2: Aspectual Categories

events have defined beginnings and ends, whereas a *state* is a state of affairs which holds true for some indefinite amount of time. The dimension of *consequentiality* indicates whether or not the event is accompanied by a transition to a state of affairs which the speaker considers to be "contingently related to other events that are under discussion." *Atomicity* indicates that the event is portrayed as punctual or instantaneous—that is, in an indivisible whole.

"Harry is at the top" is an example of a *state* from (Moens and Steedman, 1988, p. 17), because it is a situation which holds, so far as the utterance is concerned, indefinitely into the future and the past. Thus, although one might guess that Harry was not always at the top, that supposition is motivated by knowledge about *tops* (that one usually climbs to them, rather than starting there) and not knowledge about *being*.

Culminations and *Points* are the two atomic situation-types. They are distinguished by their telicity, as illustrated by the following two examples:

(a) Natasha won the race. *(Culmination)*

(b) Natasha blinked. *(Point)*

In (a) the occurrence of the event leads to a new state of affairs, in which the race is finished, and Natasha is the winner, while in (b) the event transpiring gives no new information about the state of the world.

Similarly, *Culminated Processes* and *Processes* are the two telic and atelic non-atomic situation-types.

(c) Harry climbed. *(Process)*

(d) Harry climbed to the top. *(Culminated Process)*

They are distinguished from each other because (d) leads to a new state of affairs while (c) does not; and from (a) and (b) because (c) and (d) have distinct start and an end points, while (a) and (b) have start and end points which co-occur. There is not, for example, a moment in which Natasha has begun winning the race, but has not yet finished doing so.

Moens and Steedman also note that verbs are lexically specified (possibly for more than one **aspectual type**), but that sentences can coerce a verb, so that is has a different aspectual type, by way of tenses, temporal adverbials, and aspectual auxiliaries (Moens and Steedman, 1988, p. 17).

2.3.3 Event Types

Pustejovsky (1991) proposes a classification of English verbs into one of three event types which are differentiated one from the other by their internal structure, and their relation to other events. *Transitions*, such as *give*, *open*, and *destroy*, identify an event in which a state of affairs becomes its opposite. For example, in the event described by "The door closed", the initial state of affairs is that the door is opened, and the following state of affairs is that the door is closed. The event describes the transition from one state of affairs to the next.

A *process* is a series of events denoting the same action or activity. For example, *run* denotes several instances of the action of running. The key concept is that there is more than one subevent, and further that they are the same. A *state* denotes a single event which is neither composed of subevents (like *processes*), nor evaluated relative to other

events (like *transitions*). Pustejovsky's Event Types are specified first lexically by the main verb, and second at a sentence level by the verb and other sentence constituents, in a compositional fashion.

2.3.4 Situation Types

Smith (1991) proposes a classification of verb constellations[6] into **situation types** which characterize the internal event structure of a verb, as well at its presentation. The classification is summarized in **Table 2.3**. Smith defines **situation types** as clusters of three conceptual temporal properties. These properties are *stativity*, *telicity* and *durativity*. *States*, which are the only *stative* situation types, are also the simplest. They consist of undifferentiated moments without endpoints. Much like Moens and Steedman, Smith arrives at four event situation types derived from the combinations of the *telic* and *durative* features. *Telicity* and *durativity* are analogous to Moens and Steedman's *consequentiality* and *atomicity*. In fact, *durativity* and *atomicity* are the same property with different names. Moens and Steedman define *atomicity* as portraying an event as an indivisible whole, while Smith defines *durativity* as the presence of internal stages in the temporal schema. Meanwhile, *telicity* and *consequentiality* are similar, but subtly different: *consequentiality* denotes that the event results in a meaningful change of state, whereas *telicity* indicates that the event has a goal, or natural endpoint, after which the event is complete. These two properties imply one another, but are not necessarily the same thing.

Another concept with varying terminology is the difference between lexical and phrasal aspect, which Smith refers to as *marked* and *unmarked* aspect. Smith (1991, p. 5) points out that situation type is *"signaled by the verb and its arguments"*. She later notes that events (meaning the verb and its arguments) have a conventional situation type (*unmarked*), but can be associated with a different situation type (*marked*) for emphasis, or other pragmatic reasons (Smith, 1991, p. 16).

[6]This is another term for a verb along with its accompanying complements.

Class Name	**Telic**	**Durative**	**Dynamic**
Achievements	Yes	No	No
Accomplishments	Yes	Yes	No
Semelfactives	No	No	No
Activities	No	Yes	No
States	N/A	N/A	Yes

Table 2.3: Situation Types

According to Smith, the situation type is logically independent from viewpoint, of which there are 3 possibilities:

1. *Perfective* views a situations as a whole, with start and end points.
2. *Imperfective* views less than the whole situations, specifically excluding the initial and final point.
3. *Neutral* is a flexible view which includes the initial point, and at least one internal stage.

Smith (1991, p. 10) argues that, although languages do not allow arbitrary combinations of situation type and viewpoint, an aspectual system should be general enough to capture any situation type presented as any viewpoint. By contrast, Moens and Steedman, who focus on English, have the notion of viewpoint built into the classification (1988, p. 17), presumably because in practice viewpoint and situation type are highly interdependent.[7]

2.4 Leech's Classes

All the aspectual classifications have a number of things in common. One is the top-level event/state distinction. Another is the use of atomicity and telicity (or the highly

[7]In English, at least.

Class	**Examples**
Transitional Events	hiccough, hit, jump, kick, knock, nod, tap, wink
Momentary Events	arrive, die, fall, land, leave, lose, stop
Activity	drink, eat, play, rain, read, run, talk, watch, work
Change	change, develop, grow, increase, learn, mature, widen
Perception	feel, hear, see, smell, taste, ache, feel, hurt, itch, tingle
Cognition	believe, forget, guess, think, imagine, know, suppose, understand
Attitude	hate, hope, intend, like, love, prefer, regret, want, wish
Relationship	be, belong, contain, consist, cost, depend, deserve, have, matter, own, resemble

Table 2.4: Leech's Classes

similar consequentiality) to distinguish four types of events. The distinction between lexical and phrasal aspect is another common theme. Lexical aspect is a property of verbs, whereas phrasal aspect is a syntactic construction mirroring lexical aspect (Pustejovsky, 1991). Although most classifications discuss lexical and phrasal aspect, the classifications only really work at a phrasal level, because most verbs can be compatible with multiple aspectual classes depending on the phrasal complements they are paired with. Leech, by contrast, adopts a default classification for each verb when describing his classes. He does not state his criteria for determining the default classification, so for simplicity I will assume it represents the predominant lexically specified aspect for that verb (Leech, 2004, p.23).

In other aspectual classifications all three distinguishing criteria were in some sense equally important —classes did not cross aspectual boundaries. Moens and Steedman's *culminations*, for example, are equally atomic and consequential. *Points* are as atomic as they are non-consequential. *Transition events* and *momentary events* are the analogs from Leech's classes. *Transition events* are defined primarily as consequential, with atomicity being less central. *Momentary events* are defined primarily as being non-durative, with atelicity and non-consequentiality being implied by the examples (see **Table 2.4**). The interpretation of each class with regards to the progressive gives insight into its essential nature. *Momentary events*, when combined with progressive, cannot

become extended in time, so they are interpreted as multiple events. Therefore atomicity is their essential nature. *Transition events*, on the other hand, *are* interpreted as extended when combined with the progressive, thus it is consequentiality rather than atomicity which is their essential nature. Leech, for example, notes that while *died* denotes a transition, *dying* denotes the approach to a transition. Nevertheless, *die* is assigned to the class of *transition events* because both cases involve a transition, whereas only one case involves duration.

Activities and *change verbs* are two durative event classes, as evinced by the fact that no new interpretation is required when member verbs are combined with the progressive. '*I built a house*' and '*I'm building a house*' both equally imply a certain passage of time.[8] They are distinguished one from the other by telicity. As Leech notes, *Activities* are "time-limited", meaning they must have a built-in endpoint. *Process verbs*, having non-finite but not otherwise specified duration, are atelic. As telicity or its lack is the distinguishing property between these two classes, it is also each class's primary property. Note that consequentiality is *not* a distinguishing criterion. Although most *process* verbs are non-consequential, neither is *rain*, one of the *activity* verbs.

Perception, *Cognition*, and *Attitude* verbs are three semantically-coherent groups of verbs which Leech claims are aspectually similar. Specifically, they are "unfriendly" to the progressive tense. In terms of aspectual criteria, I characterize "unfriendly to the progressive" as either stative, or atelic and non-consequential.

Delving deeper, *Perception* verbs are semantically very coherent: they denote the perception of decision-making entities. See/look, hear/listen, and feel/touch are three pairs of verbs of passive and active perception, which are respectively stative and activity-like. Other senses (taste and smell) allow active and passive uses of the same verb. One smells roses when one walks into the garden; but one also leans in to smell a bouquet of roses. Aspectually, the 'activity-like' *Perception* verbs are non-consequential, and often but not necessarily durative and non-atomic. *Looking* and *listening* are likely to be durative, but *touching* and *tasting* are more likely to be non-druative.

[8]Similarly, '*The tree grew*' and '*The tree is growing*'.

Cognition verbs, similarly, denote mental activity and have stative and non-stative senses. Verbs such as *think*, *consider* and *remember* can all be used in stative and non-stative senses.

1a) I think Mars is the fourth planet.

1b) I don't consider this a wise choice.

1c) I remember a time when people respected their elders.

2a) I thought of a way to make this work!

2b) I'm considering you for promotion.

2c) I just remembered her name, it's Suzanne.

The non-stative senses are all atelic and non-consequential, but not necessarily durative (b is durative, while a and c are not).

Finally, *Attitude* verbs denote the expression of emotion (hate, love, like) and volition (intend, want, hope, prefer), but are primarily stative. Even in traditionally non-stative tenses, such as perfect and progressive, they express facts about a person's emotions and desires rather than events.

(a) I wanted a bicycle for Christmas.

(b) I'm intending to go back to University.

(c) I hated that concert.

Relationship verbs are also almost exclusively stative, by definition: "state verbs of having or being". These verbs denotes a static relationship: abstract, temporal,or spatial.

I have referred to this dissertation as being based upon Leech's classes, and for simplicity's sake I will continue to do so, but it must be noted that in reality this dissertation is based upon my interpretation of Leech's classes. In that spirit I have taken the liberty of renaming some classes, combining others, and defining yet others more precisely. A further interpretation regards lexical aspect. As I noted earlier, Leech adopts a default classification when giving example members of each class, but otherwise exclusively discusses phrasal aspect. My final modification to Leech's classes is to interpret them as a classification of the predominate lexical aspect of verbs rather than of phrasal aspect. Therefore I will attempt to classify verbs rather than verb phrases or sentences.

2.5 Conclusion

This concludes my review of verb classification research. Although I have examined a number of aspectual classifications and just one lexical-semantic classification, the latter dominates the world of automatic verb classification. Leech's classes, which I interpret as lexical aspectual classes, fall somewhere in between standard aspectual classes and Levin's EVCA, which makes using them for automatic classification something of an untried venture.

Chapter 3

Literature Review

3.1 Chapter Overview

This chapter presents an overview of the body of research on Automatic Verb Classification. I will start by reviewing the various linguistic resources which classify verbs (Sections 3.2-3.5), then examine work in automatic classification using Levin's (1993) EVCA (Section 3.6). I conclude by examining attempts to automatically classify aspectual classes of verbs (Section 3.7).

3.2 *WordNet*

WordNet is a lexical knowledge base which is split into separate word classes following the four open syntactic categories: Nouns, Verbs, Adjectives and Adverbs. *WordNet* first splits verbs into 14 groups (Fellbaum, 1998).

1. Motion
2. Perception

3. Contact
4. Communication
5. Competition
6. Change
7. Cognition
8. Consumption
9. Creation
10. Emotion
11. Possession
12. Bodily care and functions
13. Social behavior and interactions
14. Stative

The first 13 items are semantic domains, while the last item is a catch-all group for verbs which do not fit in any of the 13 semantic domains. It includes stative verbs, as well as auxiliary and control verbs (Fellbaum, 1998, p. 70). It is worth noting that, in this case, 'stative' refers to verbs which are elaborations on the verb 'be', not the larger set of verbs which are states rather than events.

The semantic domains are used as a starting point for organizing verbs in *WordNet* into tree-like hierarchies. Nodes in the tree are synsets, groups of words which express the same (or nearly the same) concept. Each semantic field has one or more root synsets, which do not inherit from any other synset. Subordinate synsets are related to their super-ordinates by the *troponymy* relation, which means that the subordinate concept (the hyponym) is a more specific or constrained version of the super-ordinate concept (the hypernym) (Fellbaum, 1998).

Verb synsets in *WordNet* may also be related by the *Opposition*, and *Cause* relations. The *Opposition* relation refers to various types of semantic opposition, including converse and antonymy, and occurs frequently among stative and change-of-state verbs (Fellbaum, 1998, p. 82). The *Cause* relation links two verbs concepts when one causes the other; specifically, it links causative verbs to their intransitive or inchoative senses, many of

which are change verbs (Fellbaum, 1998, p. 83).

WordNet can be considered a verb classification by taking each synset as a distinct class, or by combining multiple synsets into a single class. For example, Korhonen (2002) maps *WordNet*'s verb synsets to Levin's verb classes, in order to use *WordNet* as a classification mirroring that in (Levin, 1993).

3.3 *PropBank*

PropBank builds upon the Penn Treebank (Marcus et al., 1994) by adding semantic role labels to the syntactic tags already included in the Treebank. The goal of *PropBank* is to provide a resource for semantic analysis (Kingsbury and Palmer, 2002, p. 1989). It can classify verbs using the predicate-argument structures described by the semantic role labels.

PropBank labels the expected arguments of a predicate *Arg0* to *Arg5*, and labels optional arguments as *ArgM-<tag>*, where *<tag>* is a function tag from the Treebank (Loc, Ext, Dis, Adv, Neg, Mod, Cau, Tmp, Pnc, Mnr, Dir) (Palmer et al., 2005, p. 76). The expected arguments, *Arg0* to *Arg5*, are assigned to particular argument-roles on a predicate-by-predicate basis. Predicates are defined for each major sense of a verb, so long as the senses are sufficiently distinct to have different syntactic realizations (Kingsbury and Palmer, 2002, p. 1990).

To take a couple examples from (Palmer et al., 2005, p. 75), labels for the verb "accept" are assigned as follows:

- Arg0: Acceptor
- Arg1: Thing accepted
- Arg2: Attribute

while labels for the verb "kick" are assigned like so:

- Arg0: Kicker
- Arg1: Thing kicked
- Arg2: Instrument

Note that *Arg0* is an agent-like entity in both cases, and *Arg1* is the theme and patient, respectively. This reflects the first convention of labeling, that *Arg0* should the agent, and *Arg1* should be the patient or theme. Following this convention, unaccusative verbs, like *die*, and inchoative senses of verbs, start numbering arguments from *Arg1*, not *Arg0*, because they have no agent.

The second convention is that role labelling should be consistent for members of the same *VerbNet* class (Kingsbury et al., 2002, p. 1990). Beyond that, there is no need for labels to play the same role for different predicates, even if it is for the same verb. For example, *Arg2* is the label of the **source** role for *draw* in the sense of **pull**, but is the label of the **benefactive** role for *draw* in the sense of **art** (Kingsbury and Palmer, 2002, p. 1990).

The roles assigned to a predicate are called a *roleset*. Combined with a set of associated syntactic frames allowed by the predicate in question, they form the *frameset*. All the framesets together comprise the lexicon used by the *PropBank* project.

Related frames in *PropBank* can be combined to form a *Metaframe*, which is very much like the Perspective_on relation of *FrameNet*(Ruppenhofer et al., 2010). For example, *PropBank*'s *Exchange_of_Commodities_for_cash* metaframe merges, or provides a neutral perspective on, the *buy* and *sell* frames (Kingsbury and Palmer, 2002, p. 1991).

PropBank, like *FrameNet* and *VerbNet*, is principally concerned with verbal predicate argument structures, but has the advantage of having wider coverage, with caveats, than *VerbNet*, which itself has wider coverage than *FrameNet*. The caveats are that *PropBank* covers more semantic classes than *VerbNet*, but *VerbNet* covers its classes more comprehensively (Kingsbury et al., 2002, p. 4); and that *PropBank* does not group predicates into classes of any sort.

3.4 *FrameNet*

FrameNet is a lexical resource which is not precisely a *verb* classification, because it classifies more than just verbs; however, it can be used as such by ignoring non-verb members of each group; and groups with no verbs altogether. *FrameNet* is a semantic network not of words but of frames, which are meaning-bearing structures that described situations, objects or events (Ruppenhofer et al., 2010). A semantic frame has a *Lexical Unit*, which is the word which evokes the frame, and multiple *Frame Elements*, which are the roles participants in the frame can take.

One commonly used example is the *Apply_Heat* frame, evoked by words such as *bake*, *blanch*, *boil*, *broil*, *brown*, *simmer* and *steam*; it has roles (*i.e.*, *frame elements*) of *Cook*, *Food* and *Heating_Instrument*.

This example demonstrates how *FrameNet* can act as a verb lexicon. The *Apply_Heat* frame groups 6 verbs together as highly semantically related, and gives information that applies to all of them, that they imply a situation where a cook heats food using an instrument of some kind.

The motivation of *FrameNet* is the idea that the meaning of text is best represented at a level higher than words, that words only have unambiguous meaning when they are placed in context with other words. *FrameNet* was originally conceived as a resource for computational linguistics. Therefore, evidence for the validity of a frame is based on corpus attestation (Ruppenhofer et al., 2010), and there are very specific criteria for deciding that a group of words, or other lexical units, constitute a single frame.

- The number and type of *Frame Elements* has to be the same.
- The frames must have the same entailment.
- The perspective of the concept captured by the frame must be the same.
- The lexical units must have the same relations to other frames.
- The lexical units must have the same constraints.

FrameNet attempts to enumerate valences, or semantic and syntactic combinatory possibilities (Ruppenhofer et al., 2010, p. 7), and in that regard it is very similar to *VerbNet* and *PropBank*. Although *FrameNet* has a richer set of inter-class (or inter-frame, as the case may be) relations than either *VerbNet* or *PropBank*, it suffers from a lack of coverage. *FrameNet* has 6,000 fully-annotated Lexical Units (not all of which are verbs) in almost 800 frames, with 135,000 sentences of annotated data (Ruppenhofer et al., 2010, p. 5). By contrast, *VerbNet* covers 4,536 senses verbs in 237 first-classes, and *PropBank* covers more than 4,500 framesets (analogous, for the sake of comparing coverage, to lexical units and verb senses), and has the entire Wall Street Journal Corpus as annotated data.

3.4.1 Automatic Verb Classification and *FrameNet*

Green and Dorr (2004) attempts to automatically induce frame semantic verb classes (SemFrame frames) using *WordNet* and the Longman Dictionary of Contemporary English (LDOCE) (Procter, 1978). They do so using a clustering algorithm to extract pairs of LDOCE and *WordNet* verb senses that potentially evoke the same frame. This procedure favours recall over precision by over-generating pairs, relying on a later stage to improve precision. Next they map *WordNet* synsets to LDOCE verb senses relying on matches between words:

- within the synset,
- within the definition or gloss (for LDOCE and *WordNet*, respectively),
- within example sentences,
- within word stems.

These verb sense pairs are subsequently merged into a single dataset using stringent criteria for retaining verb sense pairs, in order to favour precision and "correct" the over-generation of the first stage. The verb sense pairs form a graph, the edges of which are

generated by relating verb senses defined in terms of the same verb, relating verb senses that share a common stem, and directly extracting relationships defined by LDOCE and *WordNet*.

The graph of verb senses is input to a clustering algorithm, and the resulting clusters are hypothesized to evoke the same semantic frame, thereby constituting a *frameset*. Green and Dorr evaluate the performance of their algorithm using precision and recall, which they calculate using the so-called *majority-related* criterion. The criterion considers a SemFrame frame a match for a *FrameNet* frame if half or more of the verbs from each are semantically related. Verbs are considered semantically related if there is any overlap between the two sets of synsets generated from either frame. These synset sets are generated by associating each SemFrame and *FrameNet* verb to the *WordNet* synsets it occurs in, and adding all synsets to which the initial synsets are directly related. Using this rather forgiving majority-related criterion, the authors report recall of 83.2% and precision of 73.8%. They do not report on any more stringent measure of recall and precision.

More recently, Schulte Im Walde (2006) attempted to use human verb associations to improve feature selection. Human verb associations, in this experiment, refers to words that human volunteers suggested as related to one of 350 verbs.[1] The results of the experiment demonstrated that the verb associations did not improve clustering performance when used in conjunction with grammar-based features (e.g., subject, object, direct object, prepositional phrase), but did help when using a 20-word co-occurrence window. The experiment also demonstrated that adverbs were more informative than expected.

[1]The data was gathered using a Web-based experiment in which the subjects had 30 seconds to type as many words as they could think of which were associated to a target verb.

3.5 *VerbNet*

VerbNet is an electronic verb lexicon based on Levin's (1993) EVCA. The classes used in *VerbNet* are a modified version of the original EVCA classes, with two extensions. The first extension is based on Intersective Levin classes, in which new classes, consisting of verbs which appear in more than one of the EVCA classes, are created, eliminating overlap between classes (Dang et al., 1998). The EVCA form the top level of the *VerbNet* class hierarchy, while the subclasses are the Intersective Levin classes. Subclasses inherit all the information of their parent class, while adding additional information, such as semantic predicate information for each class (Dang et al., 2000), and information about the thematic roles and selectional restrictions which are relevant to members of the class (Dang et al., 2000). The new classes are created in order to keep all classes homogeneous with respect to all the information *VerbNet* provides, much of which was not considered when creating the original classes (Kingsbury and Kipper, 2003). The second extension incorporates 57 new classes, based on 106 new diathesis alternations (Korhonen and Briscoe, 2004).

3.5.1 Automatic Classification and *VerbNet*

One of the benefits of having so many resources with overlapping scope is that they can be used to augment and improve one another. For example, Kingsbury and Kipper (2003) describe an experiment using *PropBank* and *VerbNet*. They cluster verbs semantically using the information in *PropBank*, and compare the results to the *VerbNet* class hierarchy. The results of the experiment then suggest new members for *VerbNet* classes.

The experiment uses the annotated sentences from *PropBank*, producing 921 verbs with 200 syntactic realizations. Each syntactic realization is a list of the form (*Arg0*, *rel*, *Arg1*, *Arg2*, ...), where *rel* is a placeholder for the verb, and *Arg0*, *Arg1*, and *Arg2* denote that arguments with those labels appeared in those positions around the verb. Each verb is described as a frequency distribution over the set of all possible syntactic realizations, and clustered according to that distribution. That is, verbs which appear

with the same realizations with similar frequencies are considered similar, and therefore clustered together (Kingsbury and Kipper, 2003).

The experiment aims to compare the clusters to *VerbNet* classes, using a similarity measure, thus the number of clusters generated, which is an input to the clustering algorithm used, will affect the performance of the algorithm (Kingsbury and Kipper, 2003). Kingsbury and Kipper run their algorithm using numbers of clusters between 3 and 150, since the verbs being analyzed come from 150 *VerbNet* classes. The performance of the algorithm has local maxima at 14, 32, and 90 clusters, with 14 having the highest performance. With large numbers of clusters, however, many clusters have few verbs, while a few clusters have very many verbs. With 90 clusters total, 34 had 2 or fewer verbs, while the two largest clusters had 145 and 146 verbs. The cluster with 146 verbs roughly corresponds to the *Characterize* class within *VerbNet*, which does not have so many members. This gravitation towards a few very large classes indicate that the system is not learning Levin's classes.

Kingsbury and Kipper conclude by noting that this algorithm could be improved by adding more information to the clustering algorithm, such as the preposition (if any) which links each argument to the verb, or by using thematic roles instead of the generic labels used by *PropBank*. Finally, Kingsbury and Kipper note that the set of clusters using the clustering algorithm could be initialised using high-level *WordNet* synsets.

More recently, Brown et al. (2011) attempt to classify *VerbNet* tokens. That is, given a verb that belongs to multiple *VerbNet* classes, the classifier decides which class in particular is evoked by this particular instance of the verb. In this way, the task is one of word sense disambiguation (WSD) as much as it is verb classifier. Accordingly, the authors train an SVM classifier using typical lexical and syntactic WSD features. The lexical features are all open-class words from the target sentence and its surrounding sentences, as well as the two words to either side of the target, along with their POS tags. The syntactic features include the path through the parse tree from the target verb to its arguments, the presence of subordinate clauses and prepositional phrase adjuncts, the voice of the verb, the subcategorization frame, and the head word along with POS tag of the subject and object.

Because a verb's class membership is determined partially by its meaning, Brown et al. also include some semantic features: from each of the target's arguments, *WordNet* hypernyms, *WordNet* synonyms and named-entity tags, as well as dynamic dependency neighbours.[2] The baseline for this task is the majority *VerbNet* class, 77.78% on average. The classifier achieved 88.67% accuracy on average, which represents an error reduction of 49% over the baseline.

Finally, the authors assess the relative contribution of various groups of features by re-running the classifier with different combinations of features. They confirm that lexical features (considered standard in most supervised WSD systems) contribute the most to the performance, and further that the system performed better using all but the semantic features than when using all features (although the performance difference was negligible: 84.89% versus 84.65%).

3.6 Automatic Verb Classification - EVCA

No set of verb classes has received as much attention from automatic verb classification researchers as Levin's English Verb Classes and Alternations. Researchers have deployed supervised and unsupervised methods to her classes, and have also extended her classes to other languages. Furthermore, *VerbNet* is based on the EVCA, so in some ways the work on automatic classification and *VerbNet* is also an attempt to recognize Levin's classes.

3.6.1 Semi-Automatic Classification

Dorr (1997b) describes a method to automatically assign verbs to EVCA's semantic classes, using machine-readable dictionaries. It uses *WordNet* and the Longman Dictionary of Contemporary English (LDOCE) (Procter, 1978). The algorithm finds synonyms

[2]The dynamic dependency neighbours of a verb are other verbs which take the same nominal in the same syntactic slot—the same noun as object, for example. *cf.* Dligach and Palmer (2008)

of the target verb using *WordNet*, then uses them to generate a list of "candidate" EVCA classes (*i.e.*, by selecting all classes which contain one of the synonym verbs). Syntactic information from LDOCE is then used to select the EVCA class which most closely matches the syntax of the target verb.

Korhonen (2002) also describes a method to automatically assign verbs to Levin classes using *WordNet* in a method with some similarity to (Dorr, 1997b). The key difference, though, is that Korhonen (2002) first assigns *WordNet* synsets to EVCA classes, then uses that assignment to map particular verbs to EVCA classes.

Korhonen (2002) primarily describes the algorithm for assigning *WordNet* Synsets to EVCA classes, which takes advantage of *WordNet*'s hierarchical nature. The algorithm starts at the top-level *WordNet* synsets, and determines if they can be classified directly as an EVCA class (if the majority of a synset's members appear in the same Levin class, then it can). If so, all of that synset's subordinates (*i.e.*, its hyponym synsets) also receive that classification. The full algorithm is described in (Korhonen, 2002) as follows:

Step 1: if the majority of member verbs of a given synset S are Levin verbs from the same class, classify S directly.

Step 2: Otherwise, classify more member verbs (according to Step 4a-d) until the majority are classified, then go back to Step 1.

Step 3: Otherwise, if the classified verbs point to different Levin classes, examine whether S consists of hyponym synsets. If not, assign S to the Levin class supported by the highest number of classified verbs. If yes, go one level down in the hierarchy and classify the hyponym synsets separately, starting from Step 1.

Step 4: If S includes no Levin verbs, proceed as follows to classify the majority of member verbs of S:

(a) Extract the predominant sense of a given verb V from *WordNet*.

(b) Extract the syntactic codes from LDOCE relevant to this sense.

(c) Examine whether V could be assigned to a Levin class already associated with the other verbs in the (i) same synset, (ii) possible hypernym synset

or (iii) possible sister synsets; do it by comparing the LDOCE codes of the sense and Dorr's (1997b) LDOCE codes of the respective Levin class(es). Given the hypothesised classes, make the final class assignment manually.

(d) If no suitable class is found, re-examine the case after more verbs have been analyzed. If the classification remains unsolved, set V aside for later examination.

3.6.2 Unsupervised Classification of EVCA

Some recent work has also attempted to classify verbs of the EVCA using unsupervised machine learning. Schulte Im Walde (2000) clusters 153 verbs from the EVCA using information about their subcategorization and selectional preferences. The selectional preferences are generated by determining which *WordNet* class the verb's nominal complements belong to. The algorithm uses 23 conceptual classes which represent *WordNet* subtrees. The conceptual classes are limited to 23 in order to facilitate generalization. The clustered verbs included highly polysemous verbs, as well as high- and low-frequency verbs.

Schulte Im Walde uses two clustering algorithms, iterative distance clustering, and unsupervised latent class analysis, using the expectation maximization algorithm. Iterative clustering tended to put all the verbs into a few large clusters, and required a second pass of clustering the verbs within each group containing more than four verbs (*i.e.*, limiting the clusters to having at most four verbs). The Latent Class analysis clustered verbs into 80 clusters (this number was arrived at by experimenting with different number of clusters).

Both algorithms are run with two different inputs. The first run only knows about the subcategorization frames, and the second run knows the subcategorization frame as well as the selectional preferences of the verbs. Both performed better without than with the selectional preferences information. Iterative distance had a precision and recall of 61% and 36% with just subcategorization frame information, and a precision and recall of 38% and 20% after adding the selectional preferences. Similarly, the Latent Classes algorithm

had 54% precision and 38% recall with just subcategorization information, and 31% precision and 31% recall after adding selectional preferences. Schulte Im Walde (2000) notes that refining the input data with selectional preferences allows the algorithms to get deeper information about the lexical semantic of classes, but degrades the overall performance due to data sparseness.

3.6.3 Supervised Classification of EVCA

There has been much work over the past decade on automatic verb classification of EVCA using supervised learning. This avenue of research started with Stevenson and Merlo (1999) looking at 20 verbs each from three classes: unergatives, unaccusatives, and object-drops. Stevenson and Merlo built a classifier using four features, the relative frequencies of the transitive frame, the active voice, the past participle, and the causative. Using 10-fold cross-validation and the C5.0 decision tree algorithm, they achieve 64.2% accuracy.

Merlo and Stevenson (2001) build upon this work, using the same three classes of verbs, but adding a feature counting the relative frequency of animate subjects with the verb. Adding this feature increased classification accuracy to 69.8%, again using 10-fold cross-validation and the C5.0 decision tree algorithm.

Joanis et al. (2006) expanded this work further still, attempting to produce a feature set which is general, broad and inexpensive. "General" means it will work with theoretically all classes of the EVCA, but at least the 14 classes essayed in the paper; "broad" means that it will make use of as many distinctions between classes as possible, and "inexpensive" means it will not require computationally expensive parse tools, nor economically expensive human annotation.

The features of the feature space form four coherent groups:

- Syntactic slots and use of pronoun,

- Slot overlap,
- Tense, Voice, and Aspect features,
- Animacy.

The Syntactic slot features count the frequency with which slots for certain verbal arguments appear with the verb. Examples of slots include object, subject and different varieties of prepositional phrase.

The Slot overlap features count the frequency of instances where the same noun lemma appears in two different syntactic slots for the same verb. The features capture the alternation behaviour which defines EVCA classes.

(a) The ice melted.

(b) The sun melted the ice.

In the above example, *the ice* appears as both subject and object of the verb *melt*. These features consider slot overlaps corresponding to alternations described in (Levin, 1993).

Tense, *Voice*, and *Aspect* features include counting use of the passive voice, counting use of different tenses, appearances with certain adverbs and modals, and occurrences of the verb as a noun or adjective. The animacy features attempt to estimate the proportion of subjects for each verb which are animate.

Joanis et al. (2006) perform some experiments with the entire feature set, and other experiments with specific feature groups, in order to get a sense of how valuable they are to the classifying task. These experiments show that the Syntactic Slots feature group is the most informative, because it alone performs almost as well as the entire feature space. Furthermore, experiments in which all but one feature group were used show that removing the Syntactic Slots feature group decreases performances the most (average decrease of more than 10%).

Li and Brew (2008) furthers research on classifying Levin's classes by examining what features work best for that task. They hypothesize that features incorporating both lexical and syntactic information work the best. To verify, they compare feature sets based on dependency relations, subcategorization frames, and word co-occurrence. They find that for Levin's classes, features based on a combination of subcategorization frames and word co-occurrences works best, but that dependency relation based feature sets also work, particularly when working with a large number of classes.

3.7 Automatic Verb Classification - Aspect

There has been less work in automatic classification of verbs using aspectual classes, but still some worth noting. Early work on Aspectual classification relied heavily on Lexical Conceptual Structures (and was not quite automatic), while later work has relied on syntactic cues.

3.7.1 Automatic Aspectual Classification with Lexical Conceptual Structures

Dorr and Olsen (1997) present an algorithm which determines the aspectual class of verbs and sentences by examining their Lexical Conceptual Structure (LCS) representations (sentences are treated as Composed LCSs). LCSs are directed graphs in which each node has certain information associated with it, including a *type*, and a *primitive*. *Type* can be any one of *Event*, *State*, *Path*, *Manner*, *Property* or *Thing*. *Primitives* can belong to either the closed class or the open class. Closed-class primitives represent structural information, while open-class primitives carry semantic content (Traum and Habash, 2000, p. 52).

This algorithm builds on previous work which:

- assigns aspectual features to Levin classes (Dorr and Olsen, 1996; Olsen, 1996),

- automatically groups verbs into Levin classes (Dorr and Jones, 1996; Dorr, 1997b),
- assigns LCS templates to each verb (Dorr, 1997a).

Dorr and Olsen use four aspectual classes, State, Activity, Accomplishment and Achievement, defined by three features, Stativity, Punctuality and Telicity.

The algorithm relies on some previous work in which 191 verb classes based on the classes of Levin's (1993) EVCA have aspectual features manually assigned to them. *March*, for example, has *durative* and *dynamic* as its features, making it an Activity (Dorr and Olsen, 1997, p. 152). Dorr and Olsen have more classes than are in EVCA because they subdivide those classes which are not aspectually homogeneous. Most aspectually heterogeneous classes combine atelic *manner* verbs with telic *result* verbs (Dorr and Olsen, 1997, p. 153).

Next, Dorr and Olsen decompose sentences into Lexical Conceptual Structures. The LCS representations of sentences are then analyzed to find aspectual markers. Dynamicity is encoded in LCSs for events, such as *rust*, *dangle* and *run*, by the primitives *go*, *act*, *stay*, *cause*, *stay* and *let*; and in LCSs for states, such as *contain* and *enclose*, by the primitives *go-ext* and *be* (Dorr and Olsen, 1997, p. 154).

Durativity is indicated by the presence of *act*, *be*, *go-ext*, *cause* and *let*. Their lack indicates the reverse.

Telicity is indicated by particular types of *Path* node, or a semantic constant in the right-most leaf-node of the argument. A *Path* can be any one of AWAY_FROM, FROM, TO, TOWARD, and VIA. AWAY_FROM and TOWARD indicate telicity. Semantic constants are open-class primitives, nodes which carry semantic content rather than structural information.

The algorithm monotonically combines all the features of the sentence, and assigns the matching class. Dorr and Olsen first used the algorithm to verify aspectual conformance of their LCS database, and then applied it to determine the aspect of composed LCSs, noting that knowing verbal and sentential aspect facilitates machine translation (Dorr

and Olsen, 1997, p. 155). Finally, Dorr and Olsen note that no work they are aware of attempts to determine aspectual information on such a scale. This work is not exactly automatic classification, since it relies on knowing which Levin class the verb belongs to.

3.7.2 Pundit System

Passonneau (1988) analyzes short messages which denote situations. Each situation is classified as a *state*, *process* or *transitional event* in order to characterize the temporal structure of the situation.

This classification follows Vendler's (1967) four-class system, with *Achievements* and *Accomplishments* collapsed into the *transitional event* class (Passonneau, 1988, p.47). These classes are merged following Dowty (1986), because the difference between them is that *Accomplishments* entail a sequence of distinct subevents, while *Achievements* do not, yet any event can be seen as a sequence of subevents, given a sufficiently fine-grained timescale. A blink, to consider a canonical punctual event, when viewed in slow-motion will have two clear stages, one in which the eyes are closing, and another in which they open again. Thus, argues Passonneau, the difference between the two is a matter of time-scale, which should not be built into the aspectual system (Passonneau, 1988, p. 48).

Passonneau only applies the classification to what she calls first-order verbs. First-order verbs denote situations, and take arguments which are concrete entities, but do not take propositional arguments. There are also second- and third-order verbs, following an analogy to first-, second-, and third-order logic. Second-order verbs have propositional arguments, about which they provide temporal information (*e.g.*, *occur*, *follow*, *happen*). Third-order verbs refer to complex situations, such as *causing* (Passonneau, 1988, p. 49).

The class of a situation type is determined based on the lexical aspect and tense of the main verb of the sentence. In particular, the situation type has the same class as the lexical aspect, unless it is the progressive tense, in which case a transition event becomes a process (Passonneau, 1988, p. 52). The Lexical aspect of the verb is determined by

the semantic decomposition of the verb. If the verb is first-order, and the semantic decomposition contains a *BECOME* operator, it is a *transition event*. Otherwise, if the semantic decomposition contains a *DO* operator, it is a *process*. Otherwise, it is a *state* (Passonneau, 1988, p. 50).

3.7.3 Automatic Stativity Classification

Brent (1991) describes an implemented algorithm for determining the stativity of verbs based on the linguistic context in which they appear. Specifically, the algorithm uses certain syntactic diagnostic tests proposed by Dowty (1979): the presence of the progressive aspect, and the presence of a rate adverb (quickly, slowly, ...) modifying the verb. The algorithm uses three statistical tests to determine which class (*i.e.*, stative or non-stative) a verb falls into.

The first test is that verbs which occur at least 0.1% of the time in the progressive or with a rate adverb, with 95% confidence, are classified as events. Verbs which do not pass the first criterion, and which also appear in the progressive with a frequency below a certain upper bound, are classified as purely stative. Finally, verbs which meet neither of the above criteria are "indeterminate".

3.7.4 Automatic Aspectual Classification

Siegel and McKeown (2000) experiment with classifiers using linguistic indicators to distinguish events from states, and telic events from atelic events. The classifiers attempt to establish the aspectual class of the verb phrase rather than the aspectual class of the verb alone. The difference is that the aspectual class of the clause is that of the main verb after being coerced by the verb's complements, which may or may not change the aspectual class.

Siegel and McKeown (2000) use 14 indicators to determine the aspectual class:

- frequency
- *not* or *never*
- temporal adverb
- no subject
- past / present participle
- duration-*in* PP
- perfect
- present tense
- progressive
- manner adverb
- evaluation adverb
- past tense
- duration-*for* adverb
- continuous adverb

The value for each indicator is the frequency with which a verb appears with that phenomenon. The paper uses the English Slot Grammar parser (McCord, 1990) to produce a parse of the corpora (medical discharge summaries for the event vs state experiment, and 10 novels for the telicity experiment), which is needed to detect the 14 indicators.

Siegel and McKeown show that manner adverb is the most important feature for identifying events, when using logistic regression, which had an average accuracy of 87.6%.

Genetic programming had an average accuracy of 91.2%, by emphasizing *duration*, *in-PP*, *progressive*, *not or never*, *past tense* and *frequency*.

Decision Tree induction (DT), which had the best performance, 93.9%, used *frequency* as the root node, to distinguish the highly common *show* as a state. This is particularly domain-dependent, because *show* is very frequently used in medical discharge summaries, as in *The patient showed symptoms of...*, and overwhelmingly as a state. This usage

pattern would not be present in other domains. Regardless, further experiments show that DT still performed well without frequency as an attribute, achieving 92.4% average accuracy.

In the telicity experiments, the CART decision tree (Breiman et al., 1984) and the logistic regression classifier both had significantly better overall accuracies (74.0% and 70.5%, respectively) than the baseline, which was 63.3% (the other classifiers also had better performance, but failed the significance test). The *perfect* indicator was among the most important features for all three telicity classifiers.

3.8 Conclusion

This concludes my review of automatic verb classification research. Although there has been much work in the field, automatic aspectual classification has lagged behind classification of Levin's classes. Levin's classes have been approached with modern methods attempting ambitious multi-class classifications, while the field of aspectual classification has yet to venture past limited-scale binary classifiers applied to restricted domains. In addition to shedding light on Leech's classes, this dissertation explores how well some of the methodology applied to Levin's classes will fare when applied to aspectual classification of similar scope.

Chapter 4

Seed Set Expansion

4.1 Chapter Overview

This dissertation aims to prove the validity of a set of classes by uncovering and making explicit the knowledge required to assign verbs to them automatically. I will use supervised machine learning to do so, but Leech's initial elaboration of the classes to be used provides only a handful of verbs for each class, too few to attempt machine learning. This chapter describes the process used to expand upon the initial set of training verbs using distributional analysis. Section 4.2 briefly discusses the literature related to Distributional Analysis and Semantic Relatedness. Section 4.3 describes the material used to expand the list of training verbs, while Section 4.4 describes the method. In Section 4.5 I analyze the results of the seed set expansion.

4.2 Distributional Analysis and Semantic Relatedness

Distributional analysis refers to methods for analyzing the meaning of words by examining the words around them. This intuition was expressed by Firth (1957) as "You shall know a word by the company it keeps". Later it was more formally stated as Harris's (1968) Distributional Hypothesis: "The meaning of entities, and the meaning of grammatical relations among them, is related to the restriction of combinations of these entities relative to other entities".

The Distributional Hypothesis was later applied by Hindle (1990), who used distributional methods to detect semantically similar nouns. Selectional restrictions constrain which nouns may appear as objects or subjects of certain verbs. For example, *wine* may be *drunk*, *produced*, and *sold*, but not *pruned* (*ibid.*). Hindle characterized nouns by their mutual information score with verbs (when appearing as either a subject or object).

Later still, Lin (1998) used distributional methods to retrieve and cluster similar words using dependency triples. Each triple is composed of a head, dependency type, and modifier. "I have a brown dog", for example, would produce triples (have, subj, I) and (have, obj, dog), among others (Lin, 1998, p. 1). Lin further used these triples to produce features for words. From the previous example, (have, obj, dog) would produce a feature for *have*, obj(dog), and a feature for *dog*, obj-of(have). Lin produced lists of similar words ranked according to their similarity score. Rather than applying a cut-off threshold, he grouped the similar words together into a tree structure, and pruned subtrees which demonstrated meaning shift (Lin, 1998, p. 3).

Distributional analysis, when used to determine semantic similarity, relies heavily on a measure of semantic relatedness, a function which takes a pair of lexical units (words, lemma, *etc.*) and returns a real number expressing how semantically close they are, where the precise meaning of 'semantically close' depends on the specific measure used.

$$MSR : L \times L \to R$$

There are two approaches to determining semantic relatedness. The first approach is pattern-based, in which pairs of lexical units are related one to the other by pre-defined relations, such as hypernymy, hyponymy, meronymy, and so on, which can be identified by patterns (*e.g.*,'X is a kind of Y') (Hearst, 1992). The most well-known application of this approach is *WordNet*.

The second approach, which is more relevant to my work, is a clustering-based approach which relies on the Distributional Hypothesis. The first step in such an approach is to collect co-occurrence data from a corpus, to create a co-occurrence matrix. In the matrix, lexical units are represented as high-dimension feature vectors, in which each feature corresponds to a context (i.e., from the set of all contexts which appear in the corpus). Thus the co-occurrence matrix is composed of rows which correspond to lexical units, and columns which correspond to contexts. The value of the matrix at $M[n_i, c_j]$ is therefore the frequency of lexical unit n_i appearing with context c_j.

There are numerous approaches to representing word contexts. One approach, used in Latent Semantic Analysis (Landauer and Dumais, 1997) is to count words appearing in the text unit (sentence, paragraph, document) as the target lexical unit. Another approach, used in Word Space (Schütze, 1993) and Hyperspace Analogue to Language (Lund and Burgess, 1996) is to count words occurring inside a text window. A third approach, which is of most relevance to me, is to count words involved with the target lexical unit through a syntactic relations (Lin, 1998; Weeds and Weir, 2005).

Once the matrix has been constructed, the feature values for each lexical unit need to be weighted. The 'default' weighting is the context's raw frequency count, but that is unlikely to be illuminating. Another possible weighting method is Pointwise Mutual Information (PMI) (Manning and Schütze, 1999).

Finally, a function for comparing feature vectors needs to be selected. The dot product is one such function. The cosine measure is a special case of the dot product in which the vectors have been normalized to unit length. Cosine measure is preferred because it does not favour 'long' vectors – those with fewer non-zero features, or larger values for each feature.

4.3 Materials

I used two datasets for my distributional analysis. The first dataset is 200,000 articles from English Wikipedia recorded on October 11, 2010 (to which I will hereafter refer as *Enwiki*), which were processed using Link Grammar (Grinberg et al., 1995) and Relex (Ross et al., 2011), resulting in 4,276,598 parsed sentences. The second dataset is a parse of part of the British National Corpus (*BNC*) (BNC, 2001), also parsed using Link Grammar and Relex, resulting in 1,202,136 parsed sentences. The reason I used *Enwiki* is that the articles were pre-parsed and made freely available (Vepstas, 2011). This is significant because it took me weeks to parse a quarter as many sentences from the BNC. I wanted to use the BNC data, and went to the effort of parsing some of it, because it provides a greater variety of English, and because other work in automatic verb classifications has used it.

Link Grammar is a syntactic parser of English based on link grammars. Link Grammar represents the words in a sentence as a graph with words as nodes and labeled edges denoting the relations among words. At first, the graph is a simple list of words, and the only edges linking words are 'left' and 'right' edges, denoting the order of the words. Link Grammar transforms this graph into one with edges denoting syntactical relations between words. Relex is a dependency relationship extractor which converts the output of Link Grammar from parse trees to a set of dependency relationships identifying subject, object, direct object, and other relationships. Relex is essentially a rule engine, applying rules written by linguists to transform the Link Grammar graph—usually by adding feature labels and relationship structures, but occasionally by pruning the graph. Relex produces multiple parses, of which I keep the top four, the recommended default setting (Vepstas 2011, personal communication). Relex also identifies the tense and aspect of verbs in a sentence, from among many possibilities (see **Table 4.1**).

Other researchers attempting automatic verb classification have opted for different tools. Joanis et al. (2006) relied on a chunker (Abney, 1991) and some additional tools, while Siegel and McKeown (2000) used the English Slot Grammar parser (McCord, 1990). I use Relex because it provides a deep parse of the sentence without cobbling together the

infinitive_passive	imperative_infinitive	present_imperative
present_imperative_passive	past_perfect	present_infinitive
present	progressive	past_infinitive
future	past	past_progressive
present_future_perfect	present_future	present_future_passive
infinitive_perfect_passive	present_progressive	present_perfect
present_perfect_passive	past_passive	present_passive
infinitive_perfect	passive	past_imperative
imperative		

Table 4.1: Tenses and Aspects recognized by Relex

output of many different tools, but is also freely available to any researcher. Relex gave me everything I needed in one neat package.

4.4 Method

I constructed feature vectors for each lexical unit (in my case, verbs inflected for aspect and tense), in which each unique context (a dependency relation between the target verb and some other word) is a feature, and the value of each feature is, initially, the co-occurrence frequency. I used *SuperMatrix* (Broda and Piasecki, 2008) to do so. *SuperMatrix* is a tool for creating, storing, and manipulating co-occurrence matrices describing distribution patterns of lexical units. I chose not to include every context and every lexical unit which appears in my corpus out of concern that low-frequency contexts and lexical units would add noise to the data and worsen the results. Initial experimentation with exclusion thresholds indicated that larger thresholds led to greater average similarities, up to a limit of 500, and reduced overlap between classes.

To avoid having overly-frequent contexts drown out less frequent, but possibly more informative contexts, I used PMI score to weight the value of each feature. To compare a class to other lexical units, I needed a feature vector to represent the classes under

consideration, which I constructed by summing the feature vectors of all members of the class (and then removing those verbs from the resulting matrix). This results in a feature vector which is substantially longer than those of normal lexical unit vectors. This, however, will not affect results since I used cosine similarity to compare vectors, which ignores length, and considers only the angle between vectors.

Next, I used *SuperMatrix* to find the 100 lexical units most similar to each class. This process is repeated for both datasets (the Wikipedia articles and the BNC). I apply a stop list composed of modals, auxiliaries, and common function words mistakenly classified as verbs by Relex to each set of 800 lexical units, leaving 751 lexical units from Wikipedia and 758 from the BNC.

When producing each set of 100 lexical units, *SuperMatrix* does not know which of them have been suggested for other classes, which leads to a fair amount of overlap. Since my goal was to find more verbs that are very typical of each class, at each stage I chose to discard any verb which showed signs of ambiguity. I therefore removed all lexical units which are 'assigned' to more than one class, which leaves 480 lexical units from Wikipedia (36% attrition) and 356 lexical units from the BNC (54% attrition). I chose to discard ambiguous verbs rather than compare their similarity scores to break the tie because the fact of multiple classification shows that the lexical unit does not belong clearly enough to one class or the other, given the information available.

After discarding duplicate lexical units I attempted to combine the classifications of different tenses of the same verb, and discarded those for which there was disagreement. I discarded any verb for which different tenses have been assigned to multiple classes (e.g., $\text{fight}_{progressive}$ in one class and fight_{past} in another). This process leaves 128 verbs from Wikipedia (73% attrition) and 89 from the BNC (75% attrition). This choice, again, was meant to filter out ambiguity. If a verb is assigned to one class in one tense, and a different class in a different tense, then this may indicate that it is not so clear to which class it belongs.

Finally, I merged the two sets of verbs (i.e., from Wikipedia and the BNC), and discarded any verbs for which there was disagreement, leaving 188 verbs. Rather than

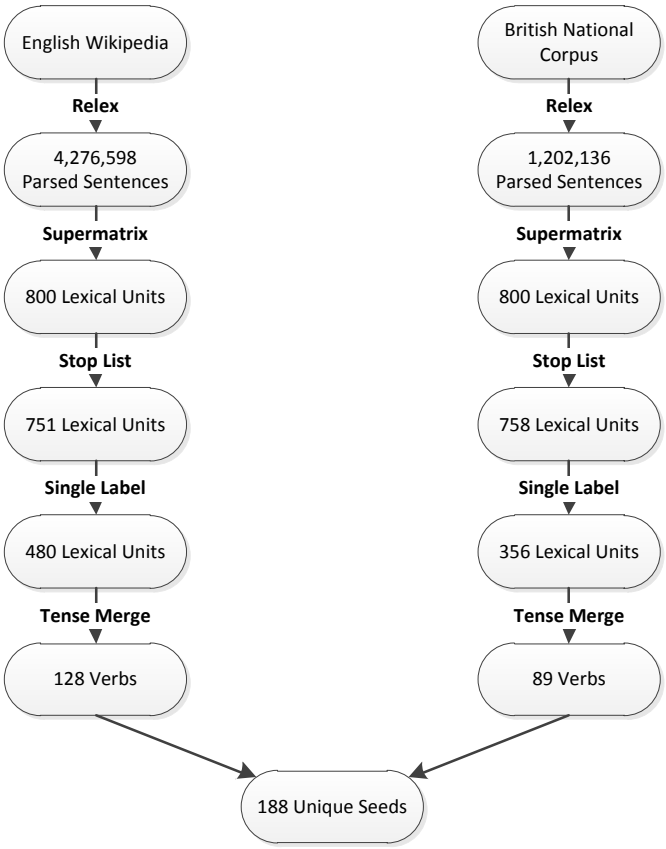

Figure 4.1: Seed filtering algorithm

keep the two lists separate and combine them at the end, I might have combined the two datasets before even starting (both are Relex parses of raw text), but I wanted to be able to compare results from each dataset to the other.

Finally, this set of verbs was combined with the gloss of their 'most frequent' sense in *WordNet*, and 9 volunteers[1] were asked to judge whether the class assignment was good or not, using the word sense from *WordNet* as a guide. The classifications are combined with *WordNet* glosses because some verbs may or may not belong in a class based on which sense of the word is being evoked. Since this system had no means of differentiating between word senses, I assume the system will assign classification based on the characteristics of the dominant sense of the word. Since the BNC has no word-sense annotations, I used *WordNet*'s sense-frequency data to approximate the most frequent senses in the BNC. To get a sense of how poor an approximation this was, I asked my judges to indicate if the classification was wrong only for the sense provided. This was the case for, on average, 7% of verbs.

4.5 Analysis

On average, evaluators said the system's classifications were correct 79% of the time (disregarding the *WordNet* glosses). The Kappa score (Fleiss') was 0.29 (Fleiss, 1971), which indicates poor agreement. Merlo and Stevenson (2000) also noted poor agreement among experts performing a verb classification for Levin's (1993) EVCA. Like them, I conclude that verb classification is a difficult task even for humans.

Furthermore, when comparing the performance of their experts against the Levin classification, Merlo and Stevenson found that the best performance was 86.5%. Similarly, the highest pair-wise agreement amongst my judges was 85.1%. Does this mean the upper bound for my classification task is 85%? Not necessarily. The judges are not themselves classifying verbs, they are rather agreeing (or not) with a pre-rendered clas-

[1] One volunteer was the author, the other 8 were all adults with University education in a variety of careers.

Agreeing Judges	#	%
9	81	43
8	114	60
7	132	70
6	142	75
5	155	82

Table 4.2: Judge-Classifier Agreement

sification. On the other hand, if, at best, two judges agree 85% of the time on whether a classification is correct, it stands to reason that they would have a similar rate of agreement when actually classifying themselves. Given that this number is close to the other upper bound for a verb classification task, and given the absence of any superior estimate of the task's maximum possible performance, I will proceed with it.

Table 4.2 shows the level of support given by human judges to the system's classifications. In 43% of cases, there was unanimous agreement amongst judges regarding the classification. 82% of cases garnered at least majority support. When analysing these results, however, one must bear in mind that the resulting verbs were filtered very stringently before coming under human scrutiny. I believe there are three factors degrading the performance of this procedure, which would be worth considering in future work.

The first factor is the lack of word-sense disambiguation. Even beyond the 7% of cases where a misclassification was due to the word-sense being wrong, WSD could improve the initial 'training data'. The verbs used to create context-vectors for each class, which are the 'training data' for this process, are quite polysemous. The average number of senses is 8.2 (likely artificially high due to 41 senses of the word *run*, and 35 sense of the word *play*), and the median number of senses is 5.5. It is possible that the context vectors of each class are being muddied by contexts belonging to a sense of one of the training words which really ought to be in a different class. For example, "*I feel a strong wind on my face.*" is an example of the Perception class, whereas "*I feel that this course of action is incorrect.*" is an example of the Cognition class. The system as it is now will incorrectly assign the contexts in the second sentence to the Perception class.

On the other hand, lack of WSD is not quite as damaging as it might at first seem. *Talk*, for example, has six senses in *WordNet*, five of which denote Activity. *Widen* has four senses in *WordNet*, all of which denote Change. A possible extension to this work would be to study how many of the original example verbs have different senses that classify differently, and how this affects resultant training of the system.

The second factor is that combining the classifications of every tense of a verb has the effect of discarding the information provided by the verb's tense, which other studies have shown to be very informative (Siegel and McKeown, 2000). The initial impetus for this decision was that a classification based on tensed verbs would be less useful for other researchers. It renders the classification incompatible with lexicons such as VerbNet and *WordNet*, none of which use tensed verbs as the basic unit. Furthermore, for my own purposes, working on tensed verbs could lead to data sparsity problems because it would split the instances of each verb among the many tenses and aspects of those verbs. It would be interesting to re-evaluate this procedure without the combination step, to see if the accuracy improves, and to what degree.

The final factor is the inherent similarity of some of the classes. Analysing the tensed verbs which were cross-classified[2] shows that 34 lexical units were assigned to both Activity and Momentary Events (representing 14 verbs), 34 were assigned to both Activity and Transition Events (representing 12 verbs), and 40 were assigned to both Transition Events and Momentary Events (representing 21 verbs). This shows that the three classes are similar to each other, and that differentiating between them will be difficult. Similarly, Cognition and Perception are two similar classes, having 52 lexical units assigned to both classes simultaneously, as are Change and Relationship, with 23 lexical units assigned to both. The similarity of Change and Relationship is more interesting than the other two cases, which are intuitive.

The distinctions between Transition Events, Momentary Events and Activities are very subtle. The distinction between Transition and Momentary events is the notion consequence; while the distinction between an Activity and any Event, is a matter of

[2]I used the verbs from the BNC for this analysis, because that corpus is a more heterogeneous sample of English.

timing–how discrete the start and end points are. The distinction between Perception and Cognition is similarly subtle, especially for an automated system. By contrast, the difference between Change and Relationship is, to a person, quite obvious. They are almost opposites, in that Relationship verbs often denote a stable situation which is precisely a lack of change. Indeed, one can see this by considering some verbs which were incorrectly classified as Change verbs: *control*, *differ*, *maintain*, *support*. These are all states which denote a lack of change.

4.6 Conclusion

This chapter reports on a method for deriving a small set of seeds from an extremely small set of seeds. Using distributional analysis I revealed 155 new seeds from the original 65 seeds.[3] 82% of the seeds suggested by the system received majority support from judges; however, low judge agreement indicates this is a task difficult even for humans, with an estimated 85% upper bound in accuracy. In the following Chapter I put the new seeds to work in machine learning experiments aimed at discovering yet more verbs belonging to Leech's classes.

[3]This is the number of seeds which garnered majority support from the judges.

Chapter 5

Machine Learning for Automatic Classification

5.1 Chapter Overview

In this chapter I present the methodology and results of machine learning experiments in automatic verb classification. The first group of experiments are aimed at refining my methodology and experiment set up, while the second group explores my feature set's performance in distinguishing Leech's classes. I find that the two Event classes are distinguished well; and that stative classes are distinguished poorly.

Section 5.2 examines the rationale and extraction of the features set, while **Section 5.3** presents a short overview of the Machine Learning algorithms with which I experimented. In **Section 5.4** I review the development of my experimental methodology, and in **Section 5.5** I analyze the main results. Finally, **Section 5.6** investigates the relative performance of each feature group.

Feature Group	**Features**
Properties of the Verb	37
Properties of Nominal Arguments	22
Properties of Prepositional Phrases	72
Properties of Adverbial modifiers	4

Table 5.1: Feature Groups

5.2 Feature Extraction

When selecting features for this experiment, I drew on previous studies in automatic verb classification, as well as linguistic literature regarding verb classes. I group the features I experiment with according to how they were extracted.

5.2.1 Properties of the Verb

Properties of the verb are so named because they deal with the verb in isolation. This group includes features for counting in which aspects and tenses the verb appears, features for counting the existence of deverbal nouns and adjectives, and features tracking whether the verb is modifiable by prefixes (*e.g.*, **be**calm and **en**slave) and suffixes (*e.g.*, item**ize** and just**ify**).

When the Relex parser (Ross et al., 2011) identifies a verb, it is tagged with one of 25 tenses recognized by the parser (**Table 4.1**). The tense and aspects features count each of these tags. The importance of verb tense has been noted by many researchers in linguistics (Leech, 2004; Pustejovsky, 1991; Nakhimovsky, 1988) and in automatic verb classification (Siegel, 1998; Joanis, 2002).

Features counting non-verbal homonyms measure how frequent nominal and adjectival forms of the verb are.[1].This pair of features was used in previous work on automatic

[1]To be precise, those are nouns or adjectives whose lemma is spelled the same as the verb's lemma,

Affix	**Example**
re-	**re**write
be-	**be**moan
en-	**en**rapture
pre-	**pre**ignite
mal-	**mal**function
-ize	incentiv**ize**
-ify	spec**ify**
-ate	enumer**ate**

Table 5.2: Affixes

verb classification (Joanis, 2002).

The final group of features relating to the verb counts co-occurrence with prefixes and suffixes, as shown in **Table 5.2**. Smith (1991) notes two aspectual phenomena arising from affixes. The first is that some affixes are productive with telic verbs. For example, *re-enable*, *rewrite*, and *re-send* all make sense because *enable*, *write*, and *send* are telic verbs; non-telic verbs are not productive with this prefix: *?rebelieve*, *?reunderstand*, *?resneeze*, *?relaugh*, *?reknock*.

The second phenomenon is verbs which arise by combining nouns and adjectives with certain affixes. *Calm* is the example I used earlier. When combined with *be*, it forms *becalm*, meaning 'to make calm', which is clearly a telic verb. In **Table 5.2** I show examples both of concatenation (*rewrite*) and derivation (*specify* comes from *specific* and *-ify*). The examples are also chosen to demonstrate that this phenomenon isn't quite as clear-cut as it might seem. ***En****able* is clearly a telic verb, but ***En****rapture* is equally not telic.

My initial attempt at capturing these features was erroneous and resulted in a value of zero for all eight features. My second attempt searched for any of the prefixes as the start of the lemma and any of the suffixes at the end of the lemma. My third attempt

e.g., (walk(v) vs walk(n) or slow(v) vs slow(a)

searched both the lemma and the token. These two attempts were fairly simplistic, and degraded the performance of the system on cross-validation.

I tried a fourth, more complicated mechanism, using a procedure where, after processing the entire corpus, each affix was removed from the start and end of each verb. If the resultant root also appeared in the corpus as a verb, noun or adjective, features on both the derived and root verb are updated. To take an example, consider the case where *rewrite* is derived from *write*. In this case, the procedure would first set a binary feature for *rewrite* indicating that it is productive with the affix *re-*; and then add the occurrence frequency of *rewrite* to a numeric feature for *write* which indicates the frequency of verbs which are derived from it (*i.e.*, *write*) using the affix *re-*.

This captures information from the concatenation case, but the derivation case is rather harder to detect. For example, *specify* is formed from the adjective *specific* and the suffix *-ify*. Some examples from my training verbs are identity/identify, collaborate/collaboration, nominate/nomin[ee|ation], indicate/indication, emigrate/emigration. Some verbs from among my training set that misleadingly fit this pattern are *hate* and *state*. Due to this complexity, I only tried detecting concatenation cases.

Since my seed verbs contain few verbs that fit the direct concatenation pattern, I manually added some in order to test this feature. To so I searched for verbs fitting the pattern with frequency greater than 1000, and used my own judgement to assign them to a class. In this manner, I added 89 new verbs. Unfortunately, this fourth attempt at affix data also degraded the system's performance on cross-validation, so I reverted to the original set and completed my analysis without affix based features.

5.2.2 Properties of Nominal Arguments

The next group of features say something about the nominal arguments of the verb: the subject, the object and the direct object. This group of features includes some which count cases where the verb is intransitive, transitive or ditransitive. These features are subject to peculiarities of Relex, which sometimes identifies an object or indirect object

without identifying a subject (or object, as the case may be). For example, given the sentence *"Several original A-1 titles succeeded and were given their own titles (...)"* the parser identifies *given* as a verb, and *titles* as the indirect object, but does not list either a subject or direct object. I therefore count a verb instantiation as ditransitive if the parser identifies an indirect object, transitive if it identifies a direct object, and intransitive otherwise.

These features also include properties of nominal arguments such as plurality, countability (count- and mass-nouns) and agency. All three of these properties are detected out-of-the-box by Relex. Smith (1991), Leech (2004) and Pustejovsky (1991) all remark on the ability of nominal arguments to modify the aspect of a verb realized in a sentence, so I include these features in order to capture a verb's affinity for being realized in particular aspects, as indicated by nominal arguments. In his work in automatic verb classification, Joanis (2002) considers the agency of noun phrases in all three syntactic slots.

I also considered features aimed at detecting selectional restrictions—perhaps by tracking the Named Entity class of nominals, or else detecting the *WordNet* synset or *Roget's Thesaurus*[2] paragraph group of the nominal, but it is not clear to me that verb classes as broad as the ones I consider would have common selectional constraints. For example, one can play an instrument or play a game. In one case the object is concrete, and in other it is an abstract concept; and that is the same verb-sense. Furthermore, as I have already noted, (Schulte Im Walde, 2000) attempted something similar while attempting to classify verbs of Levin's (1993) English Verb Classes and Alternations. She found, however, that including selection restrictions degraded the performance of the system due to data sparsity. Thus, if there is a pattern to be found in the selection restrictions on nominal arguments, recognizing the pattern would depend heavily on the coarseness of the nominal equivalence classes used. The equivalence classes would need to be coarse enough to alleviate data sparsity, but fine enough to reveal interesting patterns. An investigation to find the optimal equivalence classes is outside the scope of this dissertation, but is an interesting avenue for future work.

[2] *Roget's Thesaurus* is a machine-readable thesaurus which, similar to *WordNet*, groups nouns together by semantic similarity. *cf.* Kennedy and Szpakowicz (2008)

Completive	Durational
Locative	Instrumental
Positional	

Table 5.3: Classes of Prepositional Phrases

5.2.3 Properties of Prepositional Phrases

Using the prepositional phrase to characterize a verb has seen use in automatic verb classification (Siegel, 1998; Joanis, 2002) as well as in the linguistic literature (Smith, 1991; Pustejovsky, 1991; Nakhimovsky, 1988). Joanis counts occurrences of prepositional phrases using a specific preposition or a member of a group of prepositions. By contrast, Siegel only considers prepositional phrases using 'for' or 'in', and only those which also have a temporal component, and calls them 'durational' prepositional phrases. He is motivated in this, I believe, by Smith's (1991) prepositional phrase equivalence classes, shown in **Table 5.3**.

I follow Joanis in grouping prepositional phrases, using the same groups; although in retrospect I think following Siegel's approach might have served better.[3] By this I mean classifying prepositional phrases—most likely using hand-coded rules, but possibly with machine learning—as one of Smith's classes, and counting co-occurrence with the classes. Siegel only classified Events versus States and Transitional Event versus Momentary Event (to borrow my terminology); so he managed with recognizing just some cases of just one class. Unfortunately, any work with these classes would require first making them operational them, and second developing the classification rules (or system).

[3]I followed Joanis rather than Siegel because the former's approach is simpler and easier, and I believed it would be sufficient.

5.2.4 Properties of Adverbial modifiers

The final group of features concern adverbial modifiers to the verb, which have been linked to aspectual class by Pustejovsky (1991) and Nakhimovsky (1988). I follow Siegel, who defines the following groups of adverbs: **Temporal**, **Manner**, **Evaluation** and **Continuous**, providing a list of adverbs which belong in each group. Smith defines a different set of adverb groups: **Agentive**, **Instrumental**, **Positional** and **Durational**, but does not list their members. Both agree, though, that adverbs offer information regarding aspect. Using Smith's adverb groups might be more informative than Siegel's (which are primarily aimed at detecting duration and agency), but I ran into the same problem that Siegel's groups were ready to use, while Smith's were not.

5.2.5 Parse Errors

In **Chapter 4** I note that I chose Relex as my parser because it provided all the features I wanted from just one tool. That being said, having used Relex I have discovered flaws which would have argued against using it had I known of them ahead of time. Most problematic is Relex's poor performance with respect to identifying verbs. For example, prior to filtering, *and*, *can*, *may*, and *in* are among the twenty-most frequent verbs in my corpus.[4] I might have used the Stanford parser (Klein and Manning, 2003) instead; but that would require either abandoning the countability- and agency-related features, or else bolting some other tool onto the system in order to provide them. The Stanford parser also doesn't provide as fine-grained verb tense features. English Slot Grammar (McCord, 1990) was another potential option that has been used by automatic classification researchers, but it is not freely available to all researchers.

[4]I attempted to calculate Relex's recall and precision regarding identifying verbs in the *BNC*, but found that Relex does not split sentences the same was as the *BNC*. Relex, for example, merges all items in a list into one sentence, whereas in the *BNC*each item is treated as a separate sentence. This makes it difficult to calculate accuracy because the sentences are misaligned.

5.3 Machine Learning Approaches

In the following experiments I use the same set of parsed Wikipedia articles as were used in **Chapter 4**. I initially considered four algorithms: *kNN*, Naive Bayes, Decision Trees and SVM, and two baselines which I describe below.

I considered *kNN* because it performs well with attributes which are highly meaningful and represent significant underlying information (Sokolova 2011a, personal communication). As I developed my feature set, however, many of the deep semantic features were either discarded or did not turn out as well as I had hoped; and my features gravitated towards the shallow and simple. They form an informative picture, but only compositionally—each attribute in isolation contains very little information. This led me to consider SVM, which works well with sparse data representations (Flake, 2002). As I explain in the following section, however, *kNN* still performed well.[5] I use Decision Trees because they were used in two previous attempts at verb classification (Siegel, 1998; Joanis, 2002), and Naïve Bayes because it is quick to run and one of the "usual suspects" in Machine Learning.

I compare my results against two baselines rather than previous work because previous work variously used fewer or more classes, classes defined on different criteria, and tokens rather than types. The algorithm baseline[6] is the ZeroR classifier. It assigns every verb to the majority class. The features baseline[7] is trained on the characters in the lemmatized form of the verb. Specifically, the feature set consists of one feature for every letter of the alphabet, the value of which is the number of times that letter appeared in the verb (Sokolova 2011b, personal communication). Comparing to the algorithm baseline verifies that there is value in classifying verbs rather than choosing a class at random; and comparing to the features baseline shows that there is value in extracting the feature I chose rather then using some arbitrary, but very easy to extract, features.

[5] It ranks second best, roughly 10% better than the third-place contender, Decision Trees.

[6] So called because it is the simplest possible algorithm I could possibly use

[7] So called because it is trained on the simplest possible set of features

Denotation	#	%
$S^{(9)}$	81	43
$S^{(8)}$	114	60
$S^{(7)}$	132	70
$S^{(6)}$	142	75
$S^{(5)}$	155	82
$S^{(0)}$	188	100

Table 5.4: Seed Sets

5.4 Algorithm Selection and Tuning

While the features I used were inspired primarily by previous work in Linguistics and Automatic Verb Classification, the choice of algorithm was determined empirically.

As mentioned earlier, I initially considered four algorithms: *kNN*, Naive Bayes, Decision Trees and SVM. **Tables 5.5-5.6** show the results of experimenting with the Weka implementations of all four algorithms, using various settings and tuning parameters.

For *kNN* I ran with k=2 to k=60, and tried weighting according to the inverse of the distance, and with no weighting. I did not try weighting according to 1-distance because my dataset is not guaranteed to have distance less than one, which caused that algorithm to fail. I also experimented with different distance measures, and found that Manhattan distance worked best. I tried three variations of Naïve Bayes, one with kernel estimation, one with supervised discretization, one with neither (the default setting). I experimented with **C**, the pruning parameter, between 0.01 and 0.5 for Decision Trees. Finally, with SVM I experimented with different Kernels, but otherwise used the default settings.

I ran these algorithms with unfiltered seeds and the most strenuously filtered seeds, $S^{(9)}$—the set of seeds which at least nine (*i.e.*, all) judges agreed were classified correctly.[8]

[8]Similarly, $S^{(8)}$ denotes the set of seeds which at least eight judges agreed on, $S^{(7)}$—at least seven judges, and so on. $S^{(0)}$ denotes the unfiltered set of seeds, because "at least zero" judges agreed on the

Algorithm	**Tuning Parameters**	**F-Micro**	**F-Macro**
SVM	Normalized Polynomial Kernel	51%	47%
kNN	Inverse Weighting, k=5	39%	36%
kNN	Inverse Weighting, k=6	39%	35%
kNN	Inverse Weighting, k=11	37%	35%
kNN	Inverse Weighting, k=9	38%	34%
kNN	Inverse Weighting, k=3	38%	34%
kNN	Inverse Weighting, k=10	37%	34%
kNN	Inverse Weighting, k=7	39%	33%
kNN	No Weighting, k=6	37%	32%
SVM	Polynomial Kernel	34%	30%

Table 5.5: Comparison of Algorithms and Tuning Parameters with $S^{(0)}$

I did not split my data into a training set, development set and test set because of how little data I have. In all experiments I use ten-fold cross-validation to mitigate the effects of overfitting the training data.

Tables 5.5-5.6 show the algorithm-and-tuning-parameter combinations trained on $S^{(0)}$ and $S^{(9)}$, respectively. The column labeled F-Micro contains the micro-averaged F-Measure, while the column labeled F-Macro contains the macro-averaged F-Measure. The F-Measure is a weighted harmonic mean of precision and recall. I use the traditional F-Measure, which gives equal weight to precision and recall. The macro-average gives equal weight to each class, while the micro-average weights each class according to its size. When the class size does not reflect the likelihood of that class's members appearing in text, it is preferable to report macro average (Turney 2012, personal communication), but I report both for the sake of completeness. These tables show that SVM using a Normalized Polynomial Kernel performed better than other algorithms by a statistically significant margin. F-Micro indicates that, with the SVM classifier, $S^{(9)}$ performs better, while F-Macro indicates that $S^{(0)}$ performs better. Therefore, I will analyze the performance of different seed sets in a follow up experiment. In all, this experiment analyzed

verbs. See **Tables 5.4** for the size of each set.

Algorithm	**Tuning Parameters**	**F-Micro**	**F-Macro**
SVM	Normalized Polynomial Kernel	53%	44%
kNN	Inverse Weighting, k=4	43%	37%
kNN	Inverse Weighting, k=2	42%	36%
kNN	Inverse Weighting, k=3	42%	36%
kNN	Inverse Weighting, k=6	40%	33%
kNN	Inverse Weighting, k=7	39%	32%
kNN	No Weighting, k=3	39%	32%
kNN	Inverse Weighting, k=8	38%	31%
kNN	Inverse Weighting, k=9	37%	30%
kNN	No Weighting, k=4	37%	29%

Table 5.6: Comparison of Algorithms and Tuning Parameters with $S^{(9)}$

157 different algorithms and settings for each set of training data, but I show only the top 10, as ranked by macro-averaged F-measure.

Before analyzing seed sets, however, I decided to explore SVM's different tuning parameters. First I experimented with varying the data transformation type and the value of the complexity parameter, C, in isolation. I was going to follow up by varying both parameters simultaneously, but I found that using the non-default data transformation degraded performance so much that it did not seem worthwhile. **Table 5.7** shows the top 10 settings for SVM tuning parameters, as ranked by micro-averaged F-Measure. The default setting is *c=1.0* and to normalize the training data. Results for default **c** with standardization and with neither standardization nor normalization are also shown to illustrate why I did not investigate further tuning parameter variations. As on can see, standardization and with neither standardization nor normalization severely degrades the performance, while choice of **c** has little impact.

Earlier I compared performance on unfiltered seeds to performance on only those seeds which all 9 judges agreed were correct. I assumed that there would be a roughly linear relation between the quality of the seeds I trained on (as indicated by the degree to which judges agreed that seed was correct) and the performance of the resulting classifier.

Rank	**Complexity Parameter C**	**Filter Type**	**F-Micro**	**F-Macro**
1	c=1.0	Normalize	53%	44%
2	c=0.89	Normalize	53%	44%
3	c=0.88	Normalize	53%	44%
4	c=0.86	Normalize	53%	44%
5	c=0.85	Normalize	52%	44%
6	c=0.87	Normalize	52%	43%
7	c=0.83	Normalize	52%	43%
8	c=0.84	Normalize	52%	43%
9	c=0.82	Normalize	52%	43%
10	c=0.81	Normalize	51%	42%
68	c=1.0	Standardize	26%	18%
71	c=1.0	Neither	24%	17%

Table 5.7: Different Tuning Parameters for SVM

This is an intuitive assumption, but it bears examination, so I decided to compare the performance of different seed sets using the algorithm and tuning parameters I previously settled on.

Table 5.8 shows the micro-average F-measure of an SVM classifier trained on different seed sets, it shows that there is little variation in the average classification performance. I proceed with $S^{(7)}$ because it performs as well as any other seed set on average, and yields the best performance on *Attitude* verbs, which is the lowest performing class.

The trend for attitude in **Table 5.8**, however, is much more interesting. It peaks at 20%, then drops precipitously for 8 and 9 agreeing judges. The 19% drop in performance comes from removing **afford**, **encourage**, **permit** and **seek**. When Attitude verbs aren't classified correctly they're almost always classified as Cognition verbs, which indicates that this performance drop is caused because Attitude and Cognition are very similar semantically, and the classifier leans towards Cognition because there's more of them in the training data.

	Agreeing Judges				
	5	6	7	8	9
Activity	57%	56%	57%	60%	63%
Momentary Event	66%	67%	66%	64%	57%
Transition Event	73%	70%	71%	73%	75%
Cognition	66%	65%	64%	62%	60%
Attitude	19%	17%	20%	1%	4%
Perception	30%	31%	25%	30%	33%
Change	50%	55%	57%	61%	49%
Relationship	24%	29%	26%	21%	20%
Micro-Average	54%	55%	55%	55%	53%
Macro-Average	47.5%	48.5%	48.1%	47.0%	46.6%

Table 5.8: Training on different seed sets

5.5 Main Results

Table 5.9 shows the results of running an eight-way classifier using three algorithms, the SVM algorithm I settled upon previously, as well as two baselines I mention above. The column labeled *ZeroR* is the F-Measure of the majority class, *Activity*, while the other two columns are the F-Measures of each individual class.The entries marked with an asterisk performed better than both baselines by a statistically significant margin. **Table 5.9** shows that my feature set beats both baselines on average by a statistically significant margin. Each individual class, barring *Perception*, *Attitude* and *Relationship*, beat both baselines by a significant margin as well.

In **Table 5.9** *Events* (*Transition* and *Momentary*) and *Cognition* verbs are the top performers; *Change*, and *Activity* are mid-level performers; and *Perception*, *Relationship* and *Attitude* are the poor performers. Recalling my discussion of these classes from **Chapter 2**, the two event classes correspond to consequential and non-consequential events. *Activity* and *Change* correspond to the telic and atelic durative event classes. *Attitude* and *Relationship* verbs correspond to states; while *Cognition* and *Perception*

Class	**Class Size**	**SVM**	**SVM-Letters**	**ZeroR**
Transition Events	17.86%	71%*	24%	31%
Momentary Events	16.33%	66%*	39%	31%
Cognition	14.80%	64%*	14%	31%
Activity	18.37%	57%*	23%	31%
Change	10.71%	57%*	0%	31%
Relationship	7.14%	26%	20%	31%
Perception	6.12%	25%	0%	31%
Attitude	8.67%	19%	0%	31%
Micro-Average	**NA**	**55%***	**18%**	**31%**
Macro-Average	**NA**	**48%***	**14%**	**31%**

Table 5.9: 8-Way Classification Task Results

verbs are a mix of stative verbs and non-statives.

If one re-examines the performance groups in light of this, **Table 5.9** has instantaneous events as the top, durative events in the middle, and states at the bottom. This does not explain why Cognition performs so well, but I think that can be explained by the large numbers of Cognition verbs (relative to Attitude and Perception, at least).

Another way to look at **Table 5.9** is to see that ranking by F-Measure is very similar to ranking by distribution, but I do not think the class size is sufficient to explain the performance difference. For example, there are only eight more *Cognition* verbs than *Change* verbs, yet the classifier performs twice as well at identifying *Change* verbs.

Although I report on f-measure, the most closely comparable work in the field reports on percent accuracy. Korhonen (2010) reports that 66.3% and 58.4% are the two best accuracies for supervised automatic verb classification, on a 14-way classification task. These two papers, Li and Brew (2008) and Joanis et al. (2006), respectively, also report on an 8-way classification task, with percent accuracies of 61.7% and 66.9%. The overall accuracy of my classifier is 60%, which, while lower than the results of lexical semantic classification, is still a promising first attempt.

Class	**Size**	**F-Measure**
Transition Events	16.33%	65%
Momentary events	17.86%	56%
Cognition	14.80%	42%
Change	10.71%	39%
Activity	18.37%	36%
Perception	6.12%	25%
Relationship	7.14%	20%
Attitude	8.67%	0.00%

Table 5.10: Each class vs the rest

Class	**Size**	**SVM**	**SVM-Letters**	**ZeroR**
Transition Events	52.24%	86%*	68%	68%
Momentary Events	47.67%	85%*	68%	68%
Micro-Average	**NA**	**85%***	**68%**	**68%**
Macro-Average	**NA**	**85%***	**68%**	**68%**

Table 5.11: Momentary Events vs Transition Events

Table 5.10 summarizes how each class fared after training a binary classifier to distinguish between that class and the rest; and the relative performance of each class closely matches the pattern established by the 8-way classification. *Transition Events*, *Momentary Events*, and *Cognition* verbs occupy the top spots. *Change* and *Activity* verbs once more form the middle cohort, and *Relationship*, *Perception*, and *Attitude* verbs remain in the 'poor-performance' cohort.

I initially thought that distinguishing *Momentary* vs *Transition Events* would prove difficult, but there is a significant performance improvement over baseline (paired T-Test with 5% significance level). Furthermore, **Table 5.11** supports my hypothesis that the feature set is good at detecting consequentiality, since performance remains high even when other verb classes are taken out of the picture, and consequentiality is the distinguishing feature between the two classes in question. Siegel and McKeown (2000)

Class	**Size**	**SVM**	**SVM-Letters**	**ZeroR**
Cognition	50.00%	72%	62%	67%
Perception	20.69%	38%	2%	67%
Attitude	29.31%	20%	0%	67%
Micro-Average	**NA**	**50%**	**31%**	**67%**
Macro-Average	**NA**	**43%**	**21%**	**67%**

Table 5.12: Perception vs Cognition vs Attitude

Class	**Size**	**SVM**	**SVM-Letters**	**ZeroR**
Perception	34.29%	64%	47%	36%
Cognition	34.29%	51%	50%	36%
Attitude	31.43%	33%	23%	36%
Micro-Average	**NA**	**51%**	**40%**	**36%**
Micro-Average	**NA**	**50%**	**40%**	**36%**

Table 5.13: Perception vs Cognition vs Attitude After Re-sampling

also classified two event types, although they classified tokens, specific sentences denoting events, rather than verbs as I do. They achieved 74% overall accuracy, compared to my 86%.[9]

Table 5.12 shows the result of training a classifier to distinguish between *Perception*, *Cognition*, and *Attitude* verbs: the psychological verbs. This task did indeed prove difficult, as the classifier performs below baseline on average, and for all classes except *Cognition*. I had initially had low expectations because Leech states that the differences between these classes are primarily semantic (Leech, 2004), which I assume is harder to discern than an aspectual distinction.[10] It turns out, however, that *Attitude* verbs are pretty uniformly stative, while *Perception* and *Cognition* verbs contain a mixture of stative and non-stative verbs.

[9] **Table 5.11** shows F-Measure, not % accuracy

[10] That is because at least some aspectual signifiers are built into the surface forms of verbs.

Class	**Size**	**SVM**	**SVM-Letters**	**ZeroR**
Change	60.00%	80%	82%	75%
Relationship	40.00%	62%	50%	75%
Micro-Average	**NA**	**74%**	**70%**	**75%**
Macro-Average	**NA**	**71%**	**66%**	**75%**

Table 5.14: Change vs Relationship

Cognition verbs perform sharply better than the other two classes, and also has a markedly larger share of the training instances, which suggests that the performance difference is due to training data distribution. To investigate this conjecture, I subsampled the training data so that there was a uniform distribution of classes. The results of training on that data are shown in **Table 5.13**.

After resampling, the classifier performs about the same on average, but each individual class beats the baseline measure—albeit not by a statistically significant margin. Regardless, this result shows that the feature set is not really any better at finding *Cognition* verbs, as I conjectured earlier; *Cognition* verbs merely performed better because there were more of them in the training data.

Table 5.14 shows the result of training a binary classifier to distinguish between *Change* and *Relationship* verbs. I decided to investigate this because my judges (*cf.* **Chapter 4**) remarked that *Change* and *Relationship* verbs were often mixed up, and further that *Change* seems to be the opposite of a state, at least semantically. *Relationship* verbs denote an extended period in which there is no change; whereas *Change* verbs denote an extended period in which there is change. The two are similar in the way that hot and cold are similar: opposites, yet ontologically very close, both being temperatures.

I suspected it would be a difficult task, and this expectation was borne out. My feature set did a worse job of classifying *Change* verbs than the "letters" baseline classifier, and on the Relationship class performed worse than the random baseline. This is likely because the distinction between Change and a state is only evident at a deep semantic level.

Feature Set	**F-Micro**	**F-Macro**
All Features	55%	48%
No Adverb Features	54%	47%
Only Adverb Features	19%	14%
No Nominal Features	56%	49%
Only Nominal Features	27%	22%
No Preposition Features	48%	41%
Only Preposition Features	48%	41%
No Verb Features	48%	42%
Only Verb Features	46%	41%

Table 5.15: Feature Evaluation

5.6 Feature Evaluation

Having demonstrated that my approach to classifying Leech's classes is at least comparable to contemporary work in classifying Levin's classes, I decided to investigate the relative contribution each group of features (as summarized in **Table 5.1**) made to the overall performance of the classifier. To do so, I alternately removed each feature group from the feature set, and removed everything else but a particular group from the feature set, and re-ran the same SVM algorithm on the same seed set. The results of these experiments are summarized in **Table 5.15**.

Examining the table, we see that the adverb and nominal feature groups do not contribute much to the system's performance. F-measure only drops one point upon removing the adverb feature group, and actually goes up after removing the nominal feature group. By contrast, the preposition and verb feature groups perform as well alone as the other three feature groups combined, and almost as well as the entire combined feature set.

These two groups perform as well alone as the other three groups because the adverb features and nominal features contribute basically nothing. Therefore, the comparison

boils down to preposition features performing as well alone as verb features alone, and vice versa. This agrees with findings from Siegel and McKeown (2000); that some of the most important features for distinguishing consequential from non-consequential events are the tense, aspect, and *in*-prepositional phrases. The verb properties feature group contains features detecting tense and aspect, and the prepositions feature group will naturally detect *in*-prepositional phrases, as well as other informative prepositions, thus it makes sense that they are the two strongest groups.

That both groups perform almost as well alone as the entire feature set indicates that they have the opposite of synergy–the interaction of the two groups working in concert produces an effect barely greater than that of either group operating alone. Investigating the per-class performance of each group shows that they perform similarly well on the same classes as the full feature set. Both event classes are delineate well, as are *Change*, *Activity*, and *Cognition* verbs. The informative features from the verb feature group are *Past Perfect Tense Count*, and *Present Progressive Tense Count*; while the informative features from the preposition feature group are the prepositions *after*, *off*, *over*, and *to*. A promising avenue for future work will be to add features specifically targeted at detecting *Relationship*, *Attitude*, and *Perception* verbs.

5.6.1 Linguistic Indicators

In **Chapter 1** I state that I aim to make explicit the knowledge required to assign a verb to a class. **Chapter 2** goes a long way towards doing so by analyzing many different formulations of aspectual classes. The feature set I describe at the beginning of this section is an initial guess at the low-level facts required to delineate aspectual classes. As I discovered during the foregoing feature analysis, events are distinguished from states primarily by a handful of features: the infinitive, past progressive, and present progressive tenses; the presence of *Manner* adverbs; and numerous prepositions, such as the destination preposition group, *during*, *near*, *off*, *out of*, the source group, and *with*. Event classes (*i.e.*, including *Activities* and *Change* verbs) are distinguished one from another by the past perfect, present progressive, present perfect, and past imperative

tenses; by *Manner* adverbs; and by the prepositions *after* and *like*. Although my classifier did not perform well with stative classes, the intransitive frame and the preposition *to* were informative features for distinguishing one class from another. This collection of informative features shows that tense and prepositional phrase attachment are the primary indicators of aspectual class membership.

5.7 Applications

Many applications of an aspectual classification involve using the aspectual class of a verb to determine the aspectual class of its surrounding clause. The aspectual class of a verb clause can be determined using the aspectual class of its verb and coercion rules described, for example, by Moens and Steedman (1988).

The aspectual class of a clause can be used in Machine Translation in order to select the correct preposition in the target language. For example, *for* in English can translate to *pour*, or *pendant*, in French, as demonstrated by the following examples (Moens and Steedman, 1988).

1a) John arrived late at work for several years.

1b) Pendant des années Jean est arrive en retard au travail.

2a) John left the room for a few minutes.

2b) Jean a quitté la chambre pour quelques minutes.

The choice of preposition in French is driven by the aspectual class of the sentence. In example (1), *pendant* is selected because the clause as a whole is process, and *for* describes the length of the process. In example 2, however, the aspectual class of the clause is transitional event, and *for* describes the length of the state which obtains after the event – the amount of time during which John was not in the room. This problem is

not limited to just *for*. Many of the most common prepositions are used to convey many different meanings, and in general do not have the same set of meanings when translated to other languages.

The aspectual class of a clause has also been applied to the task of recognizing textual entailment. Recent approaches to this task have relied on determining the event structure of a sentence, and reasoning from the resultant subevents (Im, 2009).

Aspectual classes can also be of use in health surveillance and epidemiological reasoning. This is the task of analyzing reports of a disease outbreak and automatically extracting the time and location of the disease. Although simple pattern matching with a relatively small list of keywords such as 'disease' and 'outbreak' is sufficient to determine the location of many outbreaks, recent research indicates that a more sophisticated analysis of every event in a outbreak-report document is necessary in order to avoid under-reporting disease outbreaks, and to issue reports at the correct level of granularity (Chanlekha and Collier, 2010). Aspectual classes are useful in this scenario because they offer a ready-to-use list of verbs which denote kinds of events, while the specific type of event (Transitional, Momentary, Activity, Process) provides hints regarding how to analyze the event. In a similar vein, the *Attitude* and *Cognition* classes of verbs can be used in sentiment analysis to indicate sentences which are likely to denote opinions or emotions.

5.8 Conclusion

This chapter reports on supervised machine learning experiments for lexical aspectual classification. I achieve results comparable to the state-of-the-art in lexical semantic classification, and find that verb tense and prepositional attachment are the most informative features from my feature set. These features do very well at detecting aspectual consequentiality, and moderately well at detecting aspectual durativity. This leaves a performance gap waiting to be filled by features good at distinguishing between different primarily stative classes, such as *Perception* verbs, *Attitude* verbs, and *Cognition* verbs.

Chapter 6

Conclusions and Future Work

6.1 Contributions

In **Chapter 1** I stated the goal of this dissertation as making a verb class operational. I split this goal into two tasks: defining the boundaries of potential classes, and determining new class members. In **Chapters 2** and **3** I review the three different areas of work I build on: Leech's initial highly intuitive description of the classes; numerous works regarding classifying phrasal aspect; and numerous works regarding lexical semantic classes of verbs. In **Chapter 4** I automatically acquire new seeds using distributional analysis, and confirm that aspect can be treated as a lexical phenomenon rather than just a phrasal one, as demonstrated by unanimous judge agreement on over 40% of the proposed seeds. **Chapter 5** applies these seeds along with the originals in a number of supervised machine learning experiments, some of which yielded statistically significant improvements over the baseline measures. Thus I have achieved partial success at both goals.

The primary contribution of this work is laying a foundation for future work in lexical aspectual classification. It is of utmost importance that alternative approaches to research problems be attempted, even if they ultimately prove unsuccessful. The subfield

of automatic verb classification has been dominated for the past few years by a single classification, Levin's (1993)'s EVCA, which even its proponents admit has not been applied practically by researchers in other fields (Korhonen, 2010, p. 3623). One flaw with using Levin's classes as a linguistic resource is that many of the most interesting classes and alternations turn on deep semantic properties, which are difficult to extract from text. For example, there are ten kinds of "oblique" subject alternations, and the difference between them is the semantic role/noun class of the subject. The Abstract Cause Subject Alternation, structurally, is the same as the Instrument Subject Alternation, but in one the subject is an abstract cause, and the other it is an instrument.

A resource of this granularity,one which is so focused on diathesis alternations, is by no means the ideal lexical resource for every task. Nor is the state-of-the-art in NLP necessarily up to tackling such fine-grained distinctions between classes. Although Levin discusses over 100 different classes, modern work in automatically distinguishing these classes considers at most sixteen. Furthermore, some tasks, such as analysing reports of disease outbreaks in a global health surveillance system,[1] might benefit from a coarser-grained, aspect-focused classification of verbs. Other tasks might benefit from yet another classification.

This dissertation lays a foundation for future work in lexical aspectual classification by demonstrating that it is feasible. That 42% of the seeds proposed by the distributional analysis of **Chapter 4** are unanimously agreed upon by the judges demonstrates that the classes are valid—recognizable by people in general and not unique to Leech's personal experience. That some of the classification tasks outperformed their respective baselines by a statistically significant margin shows that Leech's classes are also recognizable by computer systems.

A secondary contribution of this work is verifying that the same criteria which distinguish phrasal aspect serve to distinguish lexical aspect. A final contribution worth nothing is that I have added to Leech's verbs many additional verbs which were arrived at through an objective procedure and vetted by human judges. In doing so, I showed that distributional analysis can be a useful tool in the automatic verb classification toolbox.

[1] As described in **Chapter 5**

6.2 Areas For Improvement

Although this work lays a solid foundation, there is some room for improvement. Most notably, the training data for my experiments was adversely affected by Relex's poor performance at identifying verbs. Since Relex was used in both the distributional analysis phase and the machine learning phase, the effect is felt twice—first in lowering the quality of seeds generated by the distributional analysis, and second by affecting the available training data.

A second area for improvement is in the evaluation of the machine learning experiments. A manual evaluation by human judges would have been ideal, but even having separate training, testing, and development data sets would have been preferable as well. Unfortunately, I did not have sufficient data available for the latter, and the former was not possible because I had imposed on all my available colleagues once already, so I had to rely on cross validation to evaluate my experiments.

Finally, I limited the scope of this work very early on by choosing not to consider multiple word senses. Although I think that was a necessary compromise, it is an omission which must inevitably be addressed by future work.

6.3 Future Work

As I alluded to in the previous section, there are numerous avenues for future work stemming from this dissertation. Merely re-implementing with a different parser may yield some performance gains. Furthermore, the distributional analysis phase might be improved by handling cross-class tensed verbs differently. I chose to discard verbs whose tenses ended up in different classes, but an alternative is to have each tense 'vote' on which class the verb ought to belong to, with each tense receiving a number of votes proportional to its relative frequency in the dataset.

There is much room for future work regarding the features used by the classifier.

Although I did not succeed in making use of affix-based features, I believe that if they were to incorporate more information about the derived or originating verb they might be useful. For example, rather than using a binary feature to indicate whether the verb was derived via an affix from some other verb, the feature could indicate the other verb's class, if it was known. Prepositional features could be improved by classifying attached prepositional phrases rather than merely counting the head preposition. For example, it may be possible to label prepositional phrases according to Smith's (1991) classes using a combination of head preposition and the phrases nominal or verbal constituents. In a similar vein, the classifier could benefit from a broader set of adverbial classes than the one I used. As a final note on features, some must be found which can reliably identify the stative and psychological classes, which were not classified well by my system.

Bibliography

Steven Abney. Parsing by chunks. In *Principle-based parsing*, pages 257–278. Kluwer Academic Publishers, 1991.

BNC. The British National Corpus, version 2 (BNC World), 2001.

Leo Breiman, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone. *Classification and Regression Trees*. Wadsworth, Belmont, 1984. ISBN 978-0412048418.

MR Brent. Automatic semantic classification of verbs from their syntactic contexts: an implemented classifier for stativity. In *Proceedings of the fifth conference on European chapter of the Association for Computational Linguistics*, pages 222–226, 1991.

Bartosz Broda and Maciej Piasecki. SuperMatrix: a General tool for lexical semantic knowledge acquisition. In *2008 International Multiconference on Computer Science and Information Technology*, pages 345–352. Ieee, October 2008. ISBN 978-83-60810-14-9.

Susan Brown, Dmitriy Dligach, and Martha Palmer. VerbNet class assignment as a WSD task. In *Proceedings of the Ninth International Conference on Computational Semantics*, pages 85–94. Association for Computational Linguistics, 2011. ISBN 6271234526.

Hutchatai Chanlekha and Nigel Collier. Analysis of syntactic and semantic features for fine-grained event-spatial understanding in outbreak news reports. *Journal of biomedical semantics*, 1(1):3, January 2010. ISSN 2041-1480. doi: 10.1186/2041-1480-1-3. URL `http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2895733\&tool=pmcentrez\&rendertype=abstract`.

Hoa Trang Dang, Karin Kipper, Martha Palmer, and Joseph Rosenzweig. Investigating regular sense extensions based on intersective Levin classes. In *In Proceedings of COLING-ACL98*, pages 293–299, Morristown, NJ, USA, 1998. Association for Computational Linguistics.

Hoa Trang Dang, Karin Kipper, and Martha Palmer. Integrating compositional semantics into a verb lexicon. In *Proceedings of the 18th conference on Computational linguistics*, volume 2, pages 1011–1015. Association for Computational Linguistics, 2000.

Dmitriy Dligach and Martha Palmer. Novel semantic features for verb sense disambiguation. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, pages 29–32. Association for Computational Linguistics, 2008.

B.J. Dorr. Large-scale acquisition of LCS-based lexicons for foreign language tutoring. In *Proceedings of the fifth conference on Applied natural language processing*, pages 139–146. Association for Computational Linguistics, 1997a.

B.J. Dorr and D. Jones. Role of word sense disambiguation in lexical acquisition: Predicting semantics from syntactic cues. In *Proceedings of the 16th conference on Computational linguistics*, volume 1 of *COLING '96*, pages 322–327. Association for Computational Linguistics, 1996.

B.J. Dorr and M.B. Olsen. Multilingual generation: The role of telicity in lexical choice and syntactic realization. *Machine Translation*, pages 37–74, 1996.

Bonnie J. Dorr. Large-Scale Dictionary Construction for Foreign Language Tutoring and Interlingual Machine Translation. *Machine Translation*, pages 271–322, 1997b.

Bonnie J. Dorr and Mari Broman Olsen. Deriving verbal and compositional lexical aspect for NLP applications. In *Proceedings of the 35th annual meeting on Association for Computational Linguistics*, pages 151–158, Morristown, NJ, USA, 1997. Association for Computational Linguistics.

David R Dowty. *Word meaning and Montague grammar*. D. Reidel, Dordrecht, Holland, 1979.

David R. Dowty. The effects of aspectual class on the temporal structure of discourse: semantics or pragmatics? *Linguistics and philosophy*, 9(1):37–61, 1986. ISSN 0165-0157.

Christiane Fellbaum, editor. *A Semantic Network of English Verbs*. The MIT Press, Cambridge, MA, USA, 1998.

John R Firth. A Synopsis of Linguistic Theory, 1930-1955. *Studies in Linguistic Analysis*, pages 1–32, 1957.

GW Flake. Efficient SVM regression training with SMO. *Machine Learning*, pages 271–290, 2002.

Joseph L. Fleiss. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382, 1971.

Rebecca Green and Bonnie J. Dorr. Inducing frame semantic verb classes from WordNet and LDOCE. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2004.

Denis Grinberg, John Lafferty, and Daniel Sleator. A robust parsing algorithm for link grammars. In *Proceedings of the Fourth International Workshop on Parsing Technologies*, 1995.

Z. S. Harris. *Mathematical structures of language*. Interscience Publishers, 1968.

Marti A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational linguistics - Volume 2*, volume COLING '92, pages 539—-545. Association for Computational Linguistics, 1992.

D. Hindle. Noun classification from predicate-argument structures. In *Proceedings of the 28th annual meeting on Association for Computational Linguistics*, pages 268–275. Association for Computational Linguistics, 1990.

Seohyun Im. Annotating event implicatures for textual inference tasks. *of the 5th International Conference on*, 02453, 2009. URL `http://pages.cs.brandeis.edu/~jamesp/classes/cs216-2009/readings2009/GL2009-4.pdf`.

Eric Joanis. *Automatic verb classification using a general feature space*. Master's thesis, University of Toronto, 2002.

Eric Joanis, Suzanne Stevenson, and David James. A general feature space for automatic verb classification. *Natural Language Engineering*, 14(03):337–367, December 2006. ISSN 1351-3249.

Alistair Kennedy and Stan Szpakowicz. Evaluating Roget's thesauri. In *Proceedings of ACL-08: HLT*, pages 416–424. Association for Computational Linguistics, 2008.

Anthony Kenny. *Action, Emotion, and Will*. Humanities Press, New York, 1963.

P. Kingsbury and K. Kipper. Deriving verb-meaning clusters from syntactic structure. In *Proceedings of the HLT-NAACL 2003 workshop on Text meaning*, volume 9, pages 70–77. Association for Computational Linguistics, 2003.

Paul Kingsbury and Martha Palmer. From treebank to propbank. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC-2002)*, pages 1989–1993, 2002.

Paul Kingsbury, Martha Palmer, and Mitch Marcus. Adding semantic annotation to the penn treebank. In *Proceedings of the Human Language Technology Conference*, pages 252–256, 2002.

Karin Kipper, Anna Korhonen, Neville Ryant, and Martha Palmer. A large-scale classification of english verbs. *Language Resources and Evaluation*, 42(1):21–40, 2008.

Dan Klein and Christopher D. Manning. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, volume 1, pages 423–430. Association for Computational Linguistics, 2003.

A Korhonen. Assigning verbs to semantic classes via WordNet. In *Proceedings of the 2002 workshop on Building and using semantic networks*, pages 1–7. Association for Computational Linguistics, 2002.

Anna Korhonen. Automatic lexical classification: bridging research and practice. *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, 368 (1924):3621–3632, August 2010. ISSN 1364-503X.

Anna Korhonen and Ted Briscoe. Extended lexical-semantic classification of English verbs. In *Proceedings of the HLT-NAACL Workshop on Computational Lexical Semantics - CLS '04*, pages 38–45. Association for Computational Linguistics, 2004.

Anna Korhonen, Y. Krymolowski, and Nigel Collier. Automatic classification of verbs in biomedical texts. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 345–352. Association for Computational Linguistics, 2006.

Thomas K. Landauer and Susan T. Dumais. A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, 104(2):211–240, 1997. ISSN 0033-295X.

Geoffrey Leech. *Meaning and the English Verb*. Longman, 2004.

Beth Levin. *English Verb Classes and Alternations*. The University of Chicago Press, Chicago, 1993.

Jianguo Li and Chris Brew. Which are the best features for automatic verb classification. In *Proceedings of ACL-08: HLT*, pages 434–442. Association for Computational Linguistics, 2008.

Dekang Lin. Automatic retrieval and clustering of similar words. In *Proceedings of the 17th international conference on Computational linguistics*, pages 768–774. Association for Computational Linguistics, 1998.

K. Lund and C. Burgess. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods*, 28(2):203–208, 1996.

Christopher D. Manning and H Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, Massachusetts, 1999.

Mitchell Marcus, Grace Kim, Mary Ann Marcinkiewicz, Robert Macintyre, Ann Bies, Mark Ferguson, Karen Katz, and Britta Schasberger. The Penn Treebank: Annotating Predicate Argument Structure. In *In ARPA Human Language Technology Workshop*, pages 114–119, Philadelphia, 1994. Linguistic Data Consortium.

Michael C. McCord. Slot Grammar: A System for Simpler Construction of Practical Natural Language Grammars. In *Proceedings of the International Symposium on Natural Language and Logic*, pages 118–145. Springer-Verlag, 1990.

Paola Merlo and Suzanne Stevenson. Establishing the upper-bound and inter-judge agreement in a verb classification task. In *Second International Conference on Language Resources and Evaluation (LREC-2000)*, volume 3, pages 1659–1664, 2000.

Paola Merlo and Suzanne Stevenson. Automatic Verb Classification Based on Statistical Distributions of Argument Structure. *Computational Linguistics*, 27(3):373–408, September 2001. ISSN 0891-2017. doi: 10.1162/089120101317066122.

Marc Moens and M Steedman. Temporal ontology and temporal reference. *Computational linguistics*, 14(2), 1988.

Alexander Nakhimovsky. Aspect, aspectual class, and the temporal structure of narrative. *Computational Linguistics*, 14(2):29–43, 1988.

Mari Broman Olsen. Telicity and English Verb Classes and Alternations: An Overview. Technical report, University of Maryland, 1996.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106, March 2005. ISSN 0891-2017.

Rebecca J Passonneau. A Computational Model of the Semantics of Tense and Aspect. *Computational Linguistics*, 14(2):44–60, 1988.

Paul Procter. *Longman Dictionary of Contemporary English*. Longman, London, 1978.

J Pustejovsky. The syntax of event structure. *Cognition*, 41(1-3):47–81, December 1991. ISSN 0010-0277.

Mike Ross, Linas Vepstas, and Ben Goertzel. Relex semantic relationship extractor, 2011. URL `http://opencog.org/wiki/Relex`.

Josef Ruppenhofer, Michael Ellsworth, Miriam R. L. Petruck, Christopher R. Johnson, and Jan Scheffczyk. *Framenet II: Extended theory and practice*. Berkeley, 2010.

Gilbert Ryle. *The concept of mind*. University of Chicago Press, 1949.

S. Schulte Im Walde. Can human verb associations help identify salient features for semantic verb classification? In *Proceedings of the Tenth Conference on Computational Natural Language Learning*, pages 69–76. Association for Computational Linguistics, 2006.

S. Schulte Im Walde, Christian Hying, Christian Scheible, and Helmut Schmid. Combining EM training and the MDL principle for an automatic verb classification incorporating selectional preferences. In *Proceedings of ACL-08: HLT*, pages 496–504. Association for Computational Linguistics, 2008.

Sabine Schulte Im Walde. Clustering verbs semantically according to their alternation behaviour. In *Proceedings of the 18th conference on Computational linguistics*, volume 2, pages 747–753. Association for Computational Linguistics, 2000.

H Schütze. Word space. In *Advances in Neural Information Processing Systems 5*, pages 895–902. Morgan Kaufmann, 1993.

Eric V. Siegel. *Linguistic Indicators for Language Understanding: Using machine learning methods to combine corpus-based indicators for aspectual classification of clauses.* Phd, Columbia University, 1998.

Eric V. Siegel and Kathleen R. McKeown. Learning Methods to Combine Linguistic Indicators: Improving Aspectual Classification and Revealing Linguistic Insights. *Computational Linguistics*, 26(4):595–628, December 2000. ISSN 0891-2017.

Carlota S. Smith. *The Parameter of Aspect*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1991.

Marina Sokolova and Guy Lapalme. Verbs Speak Loud: Verb Categories in Learning Polarity and Strength of Opinions. In *Proceedings of the 20th Canadian Conference on Artifical Intelligence*, LNAI, pages 320–331. Springer Berlin / Heidelberg, 2008.

Marina Sokolova and Stan Szpakowicz. Language patterns in the learning of strategies from negotiation texts. In *Proceedings of the 19th Canadian Conference on Artificial Intelligence*, LNAI, pages 288–299. Springer, 2006.

Suzanne Stevenson and Paola Merlo. Automatic verb classification using distributions of grammatical features. In *Proceedings of the ninth conference on European chapter of the Association for Computational Linguistics*, pages 45–52. Association for Computational Linguistics, 1999.

Lin Sun. Hierarchical Verb Clustering Using Graph Factorization. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1023–1033. Association for Computational Linguistics, 2011.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, volume 1, pages 173–180, Edmonton, Canada, 2003. Association for Computational Linguistics.

D. Traum and Nizar Habash. Generation from lexical conceptual structures. In *NAACL-ANLP 2000 Workshop on Applied interlinguas: practical applications of interlingual approaches to NLP*, volume 2, pages 52–59. Association for Computational Linguistics, 2000.

Zeno Vendler. *Linguistics in Philosophy*. Cornell University Press, 1967.

Linas Vepstas. Parsed English Wikipedia, 2011. URL `http://gnucash.org/linas/nlp/data/`.

Andreas Vlachos, Anna Korhonen, and Zoubin Ghahramani. Unsupervised and constrained Dirichlet process mixture models for verb clustering. In *Proceedings of the Workshop on Geometrical Models of Natural Language Semantics - GEMS '09*, pages 74–82. Association for Computational Linguistics, 2009.

Julie Weeds and David Weir. Co-occurrence Retrieval: A Flexible Framework for Lexical Distributional Similarity. *Computational Linguistics*, 31(4):439–475, December 2005. ISSN 0891-2017.