A SEMANTIC APPROACH TO AUTOMATED TEXT SENTIMENT ANALYSIS

by

Julian Brooke
B.S., Stanford University, 2001

THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF ARTS

In the
Department of Linguistics

SIMON FRASER UNIVERSITY

Spring 2009

APPROVAL

Name:	**Julian Brooke**
Degree:	**Master of Arts**
Title of Thesis:	**A Semantic Approach to Automated Text Sentiment Analysis**

Examining Committee:

Chair: **Dr. Yue Wang**
Assistant Professor of Linguistics

____

Dr. Maite Taboada
Senior Supervisor
Associate Professor of Linguistics

____

Dr. Nancy Hedberg
Supervisor
Associate Professor of Linguistics

____

Dr. Giuseppe Carenini
External Examiner
Assistant Professor of Computer Science
University of British Columbia

Date Defended/Approved: ____

ABSTRACT

The identification and characterization of evaluative stance in written language poses a unique set of cross-disciplinary challenges. Beginning with a review of relevant literature in linguistics and psychology, I trace recent interest in automated detection of author opinion in online product reviews, focusing on two main approaches: the semantic model, which is centered on deriving the semantic orientation (SO) of individual words and expressions, and machine learning classifiers, which rely on statistical information gathered from large corpora. To show the potential long-term advantages of the former, I describe the creation of an SO Calculator, highlighting relevant linguistic features such as intensification, negation, modality, and discourse structure, and devoting particular attention to the detection of genre in movie reviews, integrating machine classifier modules into my core semantic model. Finally, I discuss sentiment analysis in languages other than English, including Spanish and Chinese.

Keywords: sentiment analysis; evaluation; appraisal; semantic orientation; semantic model; genre classification

Subject Terms: Natural language processing (Computer science); Computational Linguistics; Linguistic models -- Data processing; Semantics (Philosophy); Meaning (Psychology)

ACKNOWLEDGEMENTS

I wish to thank my fellow graduate students as well as the staff and faculty in the Linguistics Department at SFU; it was a friendly and stimulating place to learn about language, with the intellectual gains stretching far beyond what is included in this thesis. Among the many good friends I found in the department, I particularly want to thank Sam Al Khatib, for always having his door open.

In terms of the content and quality of this work, the biggest thanks must go to my supervisor, Dr. Maite Taboada, first for allowing and encouraging me to pursue part of her research as my own, and second for gently guiding me each step of the way. The third member of our research group, Milan Tofiloski, also deserves my gratitude: more than once it felt like he was delaying his own thesis work to help with research related to mine. I would also like to thank the other members of my committee, Dr. Nancy Hedberg and Dr. Giuseppe Carenini, for taking the time to wade through this multi-disciplinary work, and for their insightful comments.

Finally, I want to thank my family, especially my parents, who have seen way too much of me, and my wife Hong, who has seen way too little.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Introduction

People are interested in what other people think. This somewhat obvious truth has taken on new implications with, on the one hand, the development of computing resources that both allow for essentially limitless public expression of opinion, and, on the other, an increase in computing power that facilitates the processing of vast amounts of information. In the last ten years, the interest in using computers to extract sentiment from text has gathered steam, and has already expanded into a major research project within computational linguistics, encompassing a variety of methodological approaches and potential applications. The latter include, for instance, tracking the opinions expressed in weblogs on a particular topic over time, information that would be useful to marketers, political analysts, and of course social scientists.

As with many facets of Natural Language Processing (NLP) research, there are significant challenges in teaching a computer to handle data that is distinctly human. One of necessary first steps in making opinion accessible to an automated system is some form of quantification, effectively turning fuzzy emotion into something that can be measured, calculated, and classified. The theoretical background necessary for such an transformation will be our first concern: in Chapter 1, we will examine closely past research which is relevant to this problem of opinion, including the classic psychological work of Osgood et al. (1957) and more recent linguistic taxonomies, such as the Appraisal framework (Martin and White, 2005). This discussion serves a dual purpose, introducing some of the key issues in linguistically-grounded text sentiment analysis as well as touching on some of the most basic tools to be used in the later applied research.

At present, the most popular approach to automated sentiment analysis at the level of the text involves using machine learning technology to build automated classifiers from human annotated documents. This method has shown much initial promise, particularly because it allows researchers to abstract away from the messy linguistic details, providing an impressive baseline performance in text polarity identification even with the simplest of features (Pang et al. 2002). The most obvious alternative, known as the word-counting or semantic approach (Turney 2002), involves building semantic orientation lexicons, calculating text-level sentiment based on the sentiment of words modulated by the effect of context. Chapter 2 includes a discussion of important research within these two paradigms and a critical comparison; despite the power that machine classifiers have demonstrated, I argue that some of the benefits are illusory, and that the addition of a semantic model will, in the long term, result in a more robust sentiment analysis.

Chapter 3 describes the Semantic Orientation (SO) Calculator, a software program that is designed to identify consumer reviews as positive or negative using a semantic model. The SO Calculator relies on a manually-tagged sentiment dictionary, and has various mechanisms for integrating contextual effects, including negation, intensification, and modality. The first half of the chapter is primarily concerned with theoretical questions and practical implementation, while the latter half evaluates the performance of relevant features using commonly-used corpora. Of particular interest is the performance of the SO Calculator in multi-class tasks (i.e., identifying a star rating rather than simple polarity); this type of task has only occasionally been addressed in the literature (Pang and Lee 2005), despite being uniquely reflective of opinion's gradient nature.

The use of a core semantic model does not preclude the integration of machine learning modules to boost performance by providing additional information to the SO Calculator. In Chapter 4, I explore two such options, the identification of genre at both the level of paragraph as well as at the level of text. Using a feature set based on classic explorations of text genre (Biber, 1988), I test semi- and fully-supervised methods to identify and tag paragraphs containing description or comment, discounting the former during SO calculation. I supplement this improvement with text-level sub-genre detection, an addition which optimizes the SO Calculator by allowing for the use of genre-specific configurations.

Finally, in Chapter 5, we turn our attention to sentiment analysis in languages other than English. Beginning with a review of previous research in several different languages, I present the results of two empirical studies: one is the adaptation of the SO Calculator to Spanish and a comparison, using Spanish corpora, of the new Spanish SO Calculator with both a machine learning model and the English Calculator supplemented with machine translation. The other study involves a linguistic analysis of certain sentiment-relevant features in Chinese (including unique features like sentence-final particles and compositional idioms), including a search for those features in an online forum corpus.

Sentiment analysis is a burgeoning field of study, one where a cross-disciplinary approach can result in both theoretical and practical gains. In this work, I hope to show how psychological and linguistic insights are complementary to computational resources: although the principled improvements to the model discussed below are rarely dramatic in their effects (the phenomenon, I would argue, is simply too complex), they do provide consistent performance benefits while furthering our understanding of evaluative and emotive language in general.

Chapter 1: Foundations of Sentiment Analysis

In this chapter I review in some depth several works which have influenced or are otherwise theoretically relevant to sentiment analysis; this overview is intended to serve as a basic introduction to some of the important issues in the field and motivate the applied research of later chapters. The first section discusses the semantic differential of psychologist Charles Osgood (Osgood et al., 1957), who conceptualizes evaluative polarity as a continuous numerical scale, and provides a means for determining the overall positive or negative tendency of a word or group of words. Section 2 introduces appraisal theory (Martin and White, 2005), which provides a wider framework for the identification of evaluative language, including several theoretic distinctions which have implications for computational approaches. The General Inquirer, the focus of Section 3, is a computer program that was perhaps the first attempt at automated sentiment analysis (Stone et al., 1966); some of the basic computational insights noted by the authors are still quite relevant, and the dictionaries have been a valuable resource for other researchers. Finally, I look at two recent papers: Potts (2007), who has formalized the notion of an expressive dimension independent of descriptive semantics, and Polanyi and Zaenen (2006), who discusses the various effects of context on the calculation of valence (polarity).

1.1 The Semantic Differential

In their 1957 book, *The Measurement of Meaning*, psychologists Osgood, Suci, and Tannenbaum summarize their efforts to quantify the meaning of words. As compared to referential theories of meaning usually grounded in logic, e.g., Russell (1905), semantic differential theory is both internalist and empirical; its goal is to represent the way humans conceptualize the meanings of words in terms of a vector in multidimensional "semantic space," the exact dimensions of which to be determined by experimental inquiry. What makes Osgood's theory particularly relevant to sentiment analysis is that *evaluation* (typified by the adjective pair *good-bad)* was discovered to be the single most identifiable factor contributing to word meaning; even words that were not primarily evaluative (e.g., *hot-cold*) were shown to have some loading of meaning on the evaluative axis. In the next three subsections, I will review the key methods and claims of the theory, highlighting aspects which inform the computational modeling of sentiment.

1.1.1 Mapping out Semantic Space

Osgood's model rests on (in his own words) a "tenuous but necessary assumption": that there are a finite number of "representational mediation reactions" that occur in the human brain, and that the set of reactions to a particular sign (e.g., a word) can therefore be mapped to a point in semantic space. Each type of reaction (e.g., evaluation) corresponds to an axis in this space, with the distance along the axis indicating the intensity of the reaction. For example, the sign *good* is strongly evaluative, but carries little or no other meaning, so its point in semantic space would rest on or close to the evaluative axis, a significant distance from the origin in the positive direction. *Bad* would lie the same distance in the negative direction. Mosier (1941) had previously demonstrated that evaluative words could be ranked using an 11-point scale that encompassed the semantic range from *excellent* to *awful*, but it is important to note that Osgood did not assume anything about the dimensions of semantic space (other than that they exist). Instead, he used a factorization technique to derive the dimensions. Taking advantage of the fact that any pair of polar adjectives would represent a straight line function through semantic space, Osgood asked subjects to rank individual words (e.g., *lady*) on 7-point scales created using a pair of antonyms (e.g., *good/bad*):

Lady: good ___:___:_X_:___:___:___:___bad

The location marked on the adjective scale should correspond roughly to the point on the line defined by the adjectives that is closest to the location of the given word in semantic space. When two words have no semantic dimensions in common, that point would be the origin (i.e., the middle of the scale). The worst case scenario is each antonym pair requiring its own axis, however factorization involves the minimization of the number of dimensions required to represent the given words in semantic space (this is all done computationally). With enough data, it becomes clear that, for example, the semantic dimension primarily involved in the pair *excellent*/*awful* is the same which is involved in *good/bad*, that the reason the word *masterpiece* would be ranked at the far right edge of both the *good/bad* and *excellent/awful* scales is because *good*, *excellent*, and *masterpiece* are all strongly evaluative; only one factor (one axis or scale) is needed to capture the common semantics of all three words.

A word like *lady*, however, would involve a number of other (probably more primary) semantic factors beyond the evaluative, as evidenced by a strong association with a word like *feminine*. Besides *evaluation*, Osgood identifies two other consistently relevant factors which he has classified as *potency* and *activity*. Potency is highly loaded on adjectives like *strong*, *big*, *major*, or *serious*, or nouns like *hero* or *villain* (an example of a word with positive evaluation but negative potency would be *gentle*), while words with significant activity vectors include *violent*, *blatant*, *hot,* or *virility* (*death*, on the other hand, is very inactive). In general, evaluation accounts for as much semantic variance as both potency and activity combined, and together the three factors accounted for approximately 50% of the variance seen in the data. Other, less distinct factors include *stability, novelty*, *tautness*, and *receptivity*. Osgood does not attempt to even estimate the total number of semantic dimensions, simply stating that it is probably "large." For the most part he restricts himself to these three primary dimensions, showing that they are enough to get interesting results, including clustering effects; *hero*, *virility*, and *success* are clustered together in three dimensional semantic space, as are *quicksand*, *death*, and *fate*.

The layout of "semantic space" suggests both the promise and the challenge of sentiment analysis. Language is rife with words that directly reflect the evaluative attitude of the speaker or writer, and, as Osgood's work shows, the evaluative factor seems particularly amenable to measurement and quantification. That said, any individual word may have any number of semantic components associated with it, and those other factors will have an unpredictable effect on the ultimate evaluative effect of the word. The word *fast* is not particularly evaluative, at least not intrinsically, but its particular brand of *activity* can result in a larger expression that seems to have considerable evaluative content (consider *a fast car* versus *a slow car*). In the next section I discuss congruity, which is Osgood's attempt to characterize the interaction of word meanings.

1.1.2 Shifting Towards Congruence

Having established that individual signs have associated with them a set of semantic features, each having both a polarity and an intensity, the question becomes: how do words affect each other when they co-occur? Osgood considers two archetypal relations that may be asserted between signs: association and disassociation. Associations correspond to a positive relation between two signs, for example *A is B*, *A likes B*, *A supports B*, etc. while disassociations are negative in character: *A is not B*, *A hates B*, *A avoids B*, etc. For any particular semantic dimension (e.g., evaluation), the congruity principle requires that an association asserted between two signs whose polarity and amplitude in that dimension are different will cause a shift of the meaning of each sign towards another, with the magnitude of the shift proportional to the difference between the two signs and inversely proportional to the original magnitude of the sign. Disassociation is intended to have the opposite effect, but the formula provided in Osgood et al. (1957) would lead to somewhat erratic effects, and I will not address it here. The congruity shift in the semantic value of a word w_1 with initial polarization p_1 appearing in an association with a word w_2 with initial polarization p_2 is given by the formula:

(1) $C_1 = |p_2| / (|p_1| + |p_2|) \; X \, (p_2 - p_1)$

For example, suppose the polarization of w_1 is +3 and the polarization of w_2 is -1. The above formula would yield a -1 point shift for w_1 and a +3 point shift for w_2. Note that signs with strongly positive or negative polarizations are less susceptible to shift, which seems intuitively correct. Consider a phrase like *lazy athlete*, where there is incongruence between the activity traditionally associated with being an *athlete* and the inactivity implied by the word *lazy*. Lazy seems the stronger term, and so we would tend to discount the activity level of this particular athlete rather than re-evaluating the meaning of lazy, though it is worth noting that we might have less stringent standards for classifying the laziness of an athlete as compared to the laziness of, say, a couch potato, and, as such, the meaning of lazy *does* seem susceptible to temporary shift. Osgood does not intend to suggest that the semantic vector associated with the a sign like *lazy* undergoes radical permanent change under the influence of co-occurring words, though he does indicate that these temporary movements can lead to more lasting effects; one of his experiments showed that subjects do change their general orientation towards a particular concept if the concept appears repeatedly in an associative relation with a polarized source. This is, Osgood suggests, how new concepts are learned in the first place.

Osgood provides a similar formula to derive the total polarization for a two-word combination. Again, p_1 and p_2 are the polarization for each word, and p_r is the polarization at the "point of resolution."

(2) $P_r = |p_1| / (|p_1| + |p_2|) \text{ X } p_1 + |p_2| / (|p_1| + |p_2|) \text{ X } p_2$

Perhaps the most surprising thing about this formula is that it is not at all additive when two words have the same polarity. If, for instance, w_1 has a polarization of +2 and w_2 has a polarization of +3, the polarization of the two word combination is +2.6, a result which is comparable to averaging. There is something intuitively troubling about this: if *hero* is highly positive, and *great* is highly positive, shouldn't a *great hero* be even more so? However, Osgood does not seem to be concerned with these sorts of combinations, and it is possible that his scale (which varies between +3 and -3) and his experimental methods (which rely on comparison to individual adjectives) are simply not designed to handle them. But perhaps there is validity to this sort of model, particularly in the way that an extremely positive word can be brought low by one that is merely somewhat positive: consider the undercutting feel of a "decent hero." The incongruence between the two words is a strong pragmatic signal that *hero* is not meant in a highly evaluative sense. Ideally, a computational system would do more than simply "add up" the polarity of the words, taking into account certain kinds of incongruence.

Osgood's own testing on the effect of word combination in using noun/adjective pairs (with the same basic rubric as before) casts further doubt on the validity of the formula. Though the formula applied to the *potency* and *activity* dimensions seems to correspond fairly well with his results (out of 64 items, only 7 show significant error with respect to potency and 14 to activity) , evaluation did quite poorly, with well over half (42) showing significant deviation. The deviation was generally in the direction of the adjective, and it was uniformly in the negative direction, i.e., most of the errors involved a negative adjective that had much stronger influence on the word combination than would be predicted by the formula. Osgood muses that "it is if the more unfavorable, unpleasant, or socially derogatory component were always dominant in word mixtures." This line of thinking leads us directly to the next topic, the nature of polarity itself.

1.1.3 Yang but Not Yin

Bipolarity is an essential feature of Osgood's account, and the metaphor of semantic space suggests a perfect symmetry between positive and negative (for any given dimension, half of the space is positive and half negative). Moreover, the positive axis of one dimension would be completely independent of the positive axis of another, thus there is no particular reason to equate positive *activity* with, say, positive *evaluation*; in fact, the dimensions have been chosen exactly because they *are* independent of one other. However, it is clear that this mathematically attractive conception is not psychologically valid. In *From Yang and Yin to and or but*, Osgood and Richards argue for much more general notion of positive and negative (analogous to Yang and Yin of ancient Chinese philosophy), tying together the disparate positives and negatives under two big cognitive umbrellas. Though complementary, positive and negative are not equal, a fact which is evident in their linguistic usage: Greenburg (1996) notes that positive terms are unmarked and can be used to represent the entire scale (e.g., *How tall are you*? rather than *How short are you*? for a general inquiry about height). Boucher and Osgood (1969) demonstrated that, cross-linguistically, positive adjectives appear with significantly greater frequency than negative adjectives and tend to take negative affixes rather than the other way around (*happy*/*unhappy* but *sad*/**unsad*). Negative words have special marked syntax, even in cases without explicit grammatical negation (Kilma, 1964):

(3) a. He was stupid to eat any mushrooms
b. ?? He was wise to eat any medicine
c. He seldom has any money
d. ?? He often has any money

This imbalance has implications for an automated system that intends to use polarity-carrying words as a primary means of detecting sentiment. Negative items are rarer, and thus directly comparing the number of positive and negative items in a text may lead to skewed results. On the other hand, their special morphology and syntax makes negative phrases easier to detect, allowing them to shine brightly (or rather darkly) in a haystack of default positivity.

Osgood and Richards (1973) also includes an investigation of the use of *and* and *but* in joining adjectives (*fun and exciting*, *fun but dangerous*), concluding that the use of these two connectives directly reflect the nature of bipolarity. *T*he positive connective, *and*, is used more frequently, including any case where there was no direct conflict in polarity between two items. *But* is always used when there is a polarity conflict on a particular dominant dimension (e.g., *big but delicate* for *potency*). In the case where two different dominant dimensions were in conflict, the evaluative dimension tended to be decisive, with positive evaluation corresponding to the use of *and*, and negative evaluation to the use of *but*. Taken together, these facts directly suggest techniques for automatically deriving the evaluative polarity of unknown words; we will return to this later.

To conclude, Osgood's work on the quantification of word meaning provides us with a basic framework for understanding polar language. The mathematical features of the semantic differential are such that it can be more or less directly applied in the computational realm, but perhaps more importantly the extensive experimentation carried out by Osgood and various colleagues while developing the theory indicate that sentiment models of this sort are not totally divorced from psychological reality.

1.2 The Language of Evaluation

Osgood et al (1957) noted that evaluative words tended to cluster into groups such as "morally evaluative," "aesthetically evaluative," "emotionally evaluative," and "socially evaluative," however this insight was not ultimately integrated into their core model. By contrast, the linguistic theory of appraisal presented by Martin and White in *The Language of Evaluation* (and elsewhere) contains a comprehensive classification of evaluative language, one that is not at all limited to the meanings of individual words. Basing their work in the Systemic Functional Linguistic paradigm of Halliday (2004/1994), the authors approach the question of appraisal holistically, including in their scope not only affect but also concerns such as modality, genre, and stance. In the first subsection I lay out the basic tenets of their system of attitude, and then in the second subsection I discuss the potential applicability of their theory to automated sentiment analysis.

1.2.1 An Appraisal Hierarchy

System Functional Linguistics (SFL) views language primarily as "sets of options for making meaning" (Halliday 1994: 15). Appraisal is included as one of three classes of resources for expressing interpersonal meaning, and can be further subcategorized into *Attitude*, which involves various expressions of sentiment, *Engagement*, which deals with the way a particular proposition is framed, and *Graduation*, mechanisms for scaling up or down of language, including intensification and quantification. Although *Attitude* is most obviously relevant here, all three of these sub-systems inform the problem of sentiment analysis. The Appraisal hierarchy is given in Figure 1.

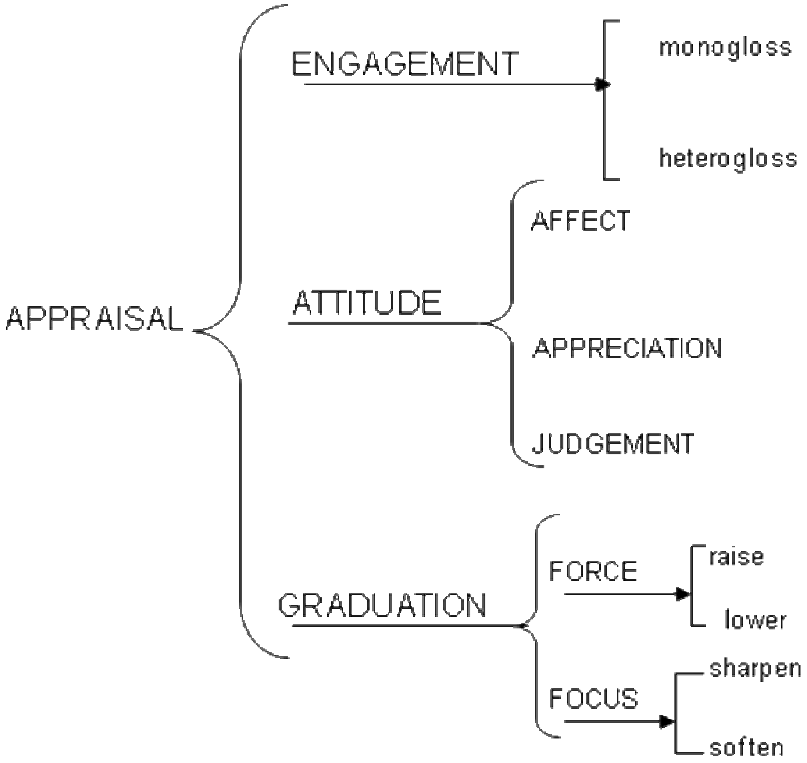

Figure 1: The Appraisal Hierarchy

Affect is the first and most primary type of attitude; it consists of emotion or behavior that directly implies emotion. Affect can be positive or negative and of varying intensity (SFL also uses a 3 point scale, low, median, and high), it can be directed or undirected, if directed the object might be realis or irrealis (e.g., *the captain feared leaving*), and it can be classified as either un/happiness (basic emotions), in/security (social well-being), or dis/satisfaction (the attainment of goals). The other two kinds of attitude are somewhat complementary; *Judgment* involves the evaluation of the behavior of agents and is classified into social esteem (normality, capacity, and tenacity, i.e., dependability) and social sanction (veracity and propriety, i.e., morality), whereas *Appreciation* is direct evaluation of objects (both physical or intangible) and events, with subcategories which include reactions of impact (e.g., *captivating*) or quality, balance or complexity of composition, and, finally, valuation, a catch-all category that includes words as diverse as *profound*, *fake*, and *helpful*. Another important distinction is between inscribed attitude and invoked attitude; the former generally consists of lexical words that are explicitly attitudinal, while the latter serves to lead the reader towards a particular pragmatic interpretation of a proposition, for instance certain common adverbs (*actually*, *only*, *however*) and lexical metaphors (*we were fenced in like sheep*).

Engagement has to do with how a speaker or writer presents propositions, i.e., as being irrefutable, open to debate, or unsupportable. Monoglossicity suggests a lack of engagement, referring to the situation where an idea is presented directly as objective fact, leaving no room for dissenting opinions. Heteroglossia, on the other hand, involves a proposition that is framed in a way that allows for alternative points of views. Heteroglossic options include: disclaming, where the textual voice is at odds with a proposition in question (*I don't believe that*…); proclaiming, where the proposition is presented as being well supported, valid, or generally agreed upon (e.g., *it is clear that…*); entertaining, where a proposition is presented as being highly subjective and susceptible to reanalysis(e.g., *I suspect that…*); and attributing, where a proposition is ascribed to someone else (the textual voice choose to distance themselves from the speech of said person, e.g., *X claims that…*).

Both of the two preceding types of attitudes are directly subject to *graduation*, the third class of attitude. There is an important distinction to be drawn between *focus graduation* and *force graduation*; focus graduation involves the sharpening or softening of a attitudinal assessment, often related to certainty or prototypicality (e.g., *a real jerk* vs. *sort of a jerk*), whereas force graduation involves the scaling up or down of sentiment, either by *intensification* of scalable adjectives, adverbs, and affect verbs (e.g., *very good*, *it slightly improves things*), or *quantification* of nouns (e.g., *a small problem*, *much joy*). Graduation is often accomplished by modification using a relatively small, closed-class set of words, however graduation is also reflected in the choice of lexical item (*good* vs. *excellent*), repetition (*hot hot hot*!), use of metaphor, and even capitalization and punctuation in text.

1.2.2 Applications

It is unlikely that we will be able to train a computer to understand the complexities of human emotive behavior in the foreseeable future, so there is naturally some question of how a detailed systematization of appraisal can be applied to automated sentiment analysis. One might argue that the classes of attitude identified in the previous section are simply either too contextual or fine-grained to be useful in anything but a complete computer model of semantics (which presumably is not on the horizon). However, insofar as the system reflects language patterns that a computer can be taught to identify, there are potential benefits to taking aspects of it into account when calculating sentiment.

Though Affect, Appreciation, and Judgment can all be used to express relevant sentiment, they differ nontrivially in their targets (e.g., *I was bored, the class was boring*, *the teacher is a bore*). In any particular text, determining whether sentiment-carrying words are directly relevant to the overall sentiment of the text is a challenging problem, but sensitivity to types of attitudes can help: for instance, if we are interested in how the author is feeling, attention could be paid to affect words that appear in the vicinity of first-person pronouns, avoiding affect elsewhere; or, in a genre such as movie reviews, our program might be more sensitive to words that signal appreciation (especially impact and quality) rather than judgment, while in genres where people are the focus (and there are no irrelevant character descriptions), the reverse might be true. In short, being aware of these distinctions is a fairly straightforward method of adapting to particular genres and the needs of the analysis (see Chapter 4).

Cues of engagement give important information about a writer's (or source's) stance towards the material they are presenting. Monoglossia is tricky; although direct statements often reflect objective facts that should not be considered in an analysis of sentiment, monoglossic evaluation is of the strongest sort (e.g., *this movie is a waste of time*). On the other hand, Heteroglossy, though often indicative of opinion, can weaken or reverse sentiment that appears elsewhere in the sentence. Attribution is particularly dicey, since the overall sentiment communicated will depend greatly on the writer's attitude towards the source, which is only sometimes hinted at (using distancing language); rather than tackling it directly, a model might simply learn to avoid basing judgments on cases when there is considerable complexity.

Compared to the other two sources of attitude, Graduation is relatively easy to integrate into an automated system, and will be discussed in detail in Chapter 3. One interesting question is whether the theoretical distinction between focus graduation and force graduation need to be handled. Focus modifiers like *really or truly* and force modifiers like *very or extremely* do not, I think, differ much in their overall effect on the intensity of the words the modify; even if *really good* means *good and I'm telling you the truth* while *very good* means *more than just your average good*, the overall cognitive result of the certainty in former case and intensification in the second case seem directly comparable (and I think people use them interchangeably). There seems little value, then, in distinguishing between the two, and for simplicity I refer to them jointly as intensification. Certain types of quantification that clearly scale up the evaluation (a *big problem* is worse than a *problem* or *a few small problems*) can also be included under the umbrella term of *intensification*, but others have a much more extreme effect on the semantics (*a lack/paucity of talent*) and thus require special consideration.

Our review of the classification of evaluative language in Martin and White (2005) has touched on a number of key issues that will be explored in more detail later on. From a theoretical standpoint, it is not enough to simply assign a positive or negative polarity to individual words; there are different types of polarized terms and other factors that are should be taken into account when interpreting them.

1.3 The General Inquirer

The first automated system for detecting use of evaluative language was built not for psychologists or linguists, but rather for social scientists. The General Inquirer (Stone et al., 1966, Stone, 1997) is a computer program originally developed in the early sixties in order to carry out content analysis, an approach to sociology that uses language as direct empirical evidence. Though the scope of the General Inquirer is quite a bit broader than sentiment analysis, many of the challenges encountered by these early researchers are still relevant to the present research; the computer hardware and computer languages have changed, but the basic properties of natural language have not. In addition, an expanded version of one of the dictionaries created for the system is still in use by many researchers today (e.g., Kennedy and Inkpen, 2006).

The General Inquirer (GI) is basically a word counter. Using dictionaries especially designed for a particular research project, the words in the text are first given tags that identify their relevant qualities, and then the GI counts the total number of instances of a tag in a text or (often) a set of texts. To give a simple example, a researcher might count the appearance of SELF tags (*I*, *me*, *myself, my, mine*) versus the appearance of SELVES tags (*we, us, ourselves, our, ours*) in a group of documents and come to a conclusion about whether the authors are oriented towards the individual or the collective. Supplemented with a dictionary of positive and negative words, the GI can be applied to text sentiment analysis. The first such dictionary was developed by a political scientist at Stanford (Holsti, 1964), based on the semantic differential framework of Osgood (Osgood et al., 1957). Originally used for tracking political trends, this dictionary has been expanded and integrated into a larger dictionary, and now contains over 4000 words that carry positive or negative sentiment[1]. It was built by hand, with each word assigned tags after a consensus was reached among three or more judges. Note that since the GI is tag based, the dictionary does not have continuous values indicating the degree of a particular tagged property; a word is positive, negative, or neutral.

Even with a dictionary already on hand, there are complexities in word counting. The original version of the GI implemented a simple syntactic marking scheme, distinguishing subjects, verbs, and objects and breaking up main and attributive clauses. A more complex version was later added (Kelly and Stone, 1975) that makes the “markers” essentially analogous to part of speech tags (in the GI, the word *tag* usually refers to content tags, e.g., *positive* or *negative*), including markers that distinguish verb tenses. Marking words for syntactic category has obvious advantages, serving as a basic kind of Word Sense Disambiguation (distinguishing, for instance

[1] Available at http://www.wjh.harvard.edu/~inquirer

kind meaning *type* from *kind* meaning *nice*). A related preliminary step in the processing of texts is the removal of inflectional endings, so that words can be properly matched to the lemmas in the dictionary without having to list all possible forms.

It is often necessary to look beyond the individual word in order to get the right meaning. The original GI included simple handing of idiomatic forms and other multiword expressions; a word might have a default tag, but would be given a different tag if it appeared in the close vicinity of another word which forms an idiomatic expression. An (archaic) example from Stone et al. (p 88):

(4) Belfry =(w, 3, BAT, 41)

This means that *belfry* will be assigned the special tag 41 (indicating, perhaps, a negative judgment instead of a physical location) if it appears within 3 words of *bat* (allowing for syntactic number of variations on the idiom *bats in the belfry*). The use of a complex tagging system allows for another kind of word sense disambiguation; rules can be written such that if *bat* appears in a sentences with NATURAL-OBJECT, it will also be tagged NATURAL-OBJECT, but otherwise a tagging of TOOL is more appropriate in certain cultural contexts.

Though positive and negative tagging seems the most obviously relevant tool for sentiment analysis, the wide range of other tags available in the GI dictionary (there are over 100) might become useful as additional techniques are developed. For instance, it is not enough to say that something that is tagged DANGER should be have a negative effect on overall text sentiment; what if DANGER appears in proximity of a word that signals AVOID or CONTROL? And even if the DANGER is unavoidable, what if it is not in the context of SELF or SELVES but a negative polarity OTHERS? In order to properly evaluate sentiment in text, it is important to capture some notion of congruence; the overall polarity of a phrase which contains a negative word depends on the nature of the target and whether the relation expressed is association or disassociation. In short, context matters, and other tags might be a good resource for determining polarity when positive or negative words are misleading or absent altogether.

1.4 The Expressive Dimension

Working within the framework of mainstream compositional semantics, Potts (2007) provides a formal definition for an *expressive* dimension which is separate from the *descriptive* dimension, namely the dimension described by predicate logic. This expansion of the theory is a response to the problem of representing expressives like *damn*:

(5) That *damn* dog kept me up all night

Potts argues that *damn* does not contribute any descriptive meaning to the sentence (the use of *damn* does not directly predicate anything about the entity *dog* in the world), but does contribute expressive meaning by limiting the interval of an expressive index. An expressive index (which itself is a kind of entity), consists of an <*a* **I** *b*> triple where *a* and *b* are entities and

I is a subinterval of [-1,1]. The interpretation of <*Bob* [-1,-0.5] *dog*>, for instance, would be that Bob (apparently) has strongly negative feelings towards the dog. Potts supposes that all entity pairs begin with a [-1,1] interval (equivalent to a clean slate), and then expressives act to progressively narrow the interval. For example, if Bob continuously swears in the context of the dog, the listener naturally become more and more certain of his strongly negative attitude towards it. The idea of a narrowing interval is quite an interesting one, and could potentially be integrated into automated sentiment analysis (see, for instance, the discussion of negation in Chapter 3). Another advantage of the theory is that it includes the notion of a source and a target as being fundamental to opinion, something which has been argued for sentiment analysis at the level of the sentence (Bloom et al., 2007).

However, Potts places a number of limits on what can be called an expressive: expressives have no descriptive content; they are applicable only to the moment of speech; they cannot be paraphrased with non-expressives to the same effect; they are tied to the speaker (when you use *damn* you can only express your own feelings, not those of others); their effects are immediate; and they can be repeated without redundancy (repetition strengthens the emotive content). Unfortunately, this severely limits the practical applicability of the theory to sentiment analysis. For instance, he would likely still handle sentences like *I hate that dog* within the descriptive dimension, e.g., hate(Bob, dog), since *hate* (and the vast majority of words which have a polarity) does not satisfy the narrow definition of an expressive. For our purposes, it would be preferable to view expressives as the extreme of evaluation (they are unambiguously and vividly evaluative), but allow for other words to have a mixture of expressive (evaluative) and descriptive content. Nevertheless, Potts' theory of expressives is promising in terms of its ability to bridge the gap between different conceptions of meaning.

1.5 Contextual Valence Shifters

Polanyi and Zaenen (2006) focus on the building of a theoretical framework for determining how context affects the polarity of a valence (polar) term. They start by assuming a numerical +2/-2 value (a valence) on positive/negative words in the lexicon (including adjectives, nouns, verbs, and adverbs), and then suggest how this numerical value should change based on the surrounding context. Negation is perhaps the clearest case of a contextual valence shifter; the authors propose that the presence of a negating word (such as *not*) should switch the sign on the valence, +2 *clever* becomes -2 *not clever*. The presence of an intensifier (*very*) or a downplayer (*somewhat*) affects the valence by increasing or decreasing the absolute value; if *nice* is +2, *somewhat nice* is +1, whereas *mean* (-2) becomes *very mean* (-3).

Less obvious valence shifters include modals and other words that express *irrealis*, where the purpose is not to describe an event that has occurred, but rather express an attitude towards a potential event. Words such as *might*, *could*, and *should* can neutralize the overall valence of a term that appears within their scope:

(6) a. Mary is mean to her dogs. She is a terrible person.
b. If Mary were mean to her dogs, she would be a terrible person.

Whereas the first sentence is highly negative of Mary, the second sentence does not claim that Mary is mean or terrible; the use of *if* and the subjunctive in the first clause, and *would* in the second clause, tell us that we are discussing a counterfactual, and that the words used should not contribute to our overall opinion of Mary. If anything, we are presupposing the opposite, though Polanyi and Zaenen do not suggest that we use *not mean* as the true valence term in this sentence, which would give it a +2 overall. Other words that carry presuppositions include *barely* and *even*; these words can often neutralize or reverse the polarity of a valence term, or carry valence even in otherwise neutral contexts (e.g., *he barely made it home* suggest something bad happened on the way) .

Discourse connectives can signal a valence shift. For instance, the authors propose that valence terms appearing in a clause with *although* (which indicates a concession) should be disregarded, since the nature of a concession relation is such that the nucleus (the other clause) is usually expressing the more primary information.

(7) Although I liked some of it, it was mostly a disappointment.

If we count both *liked* and *disappointment*, the overall valence for this sentence is 0, however it is clear that the overall attitude being expressed is negative. Polanyi and Zaenen also point out that if evidence or further elaboration is provided that has the effect of supporting a preceding positive or negative evaluation, the overall effect is strengthened.

When the purpose of the sentiment analysis is to detect the opinion of the author towards a particular topic under discussion, there are a number of potential pitfalls. First is the use of reported speech, which may present the opinion of someone else who, perhaps, the author does not agree with. Another problem is that the discussion of a topic (for instance, a movie) might be divided up into several subtopics of varying importance; the author might like the costumes, but hate the acting and the script. In these cases, it might be helpful if the analyzer had some information about the particular (sub)-genre so that only important topics and relevant statements contribute to the overall text sentiment; we will investigate this empirically in Chapter 4.

Chapter 2: Approaches to Automated Sentiment Analysis

Sentiment analysis (or *opinion mining*) has been the focus of growing attention among computational linguists in recent years, in no small part because of the emergence of the Web, which provides both a vast corpus and a variety of potential applications. The basic goal of automated sentiment analysis is the classification of language which carries an evaluative or affective stance. Esuli and Sebastiani (2005) note that this task can be divided into three interrelated subtasks: determining whether a certain unit of language is subjective, determining the orientation or polarity of subjective language, and determining the strength of that orientation. Work in sentiment analysis can be further classified according to the particular unit of language the researcher is focused on, i.e., word, phrase, sentence, or the full text. The automated system to be presented in Chapter 3 is designed to determine the evaluative polarity of an entire text, and so the present review will focus on sentiment analysis of this type, though other varieties will play a role in the discussion; for a more general review, see Pang and Lee (2008).

Research on the classification of text by polarity has been dominated by two basic approaches: one focuses on words and phrases as the bearers of *semantic orientation* or *SO* (Hatzivassiloglou and McKeown, 1997); the overall SO of the text is then simply an averaged sum of the SOs of any SO-bearing components. The most well-known example of what we will call a *semantic model* is probably Turney (2002), who used SO values automatically calculated using internet hit counts. The second approach is the *machine-learning model*, popularized by the work of Pang et al. (2002), which regards sentiment classification as simply another form of *text classification* (e.g., by topic or genre). In machine text classification, the focus is generally placed on selecting the appropriate machine-learning algorithm and the right text features, and then providing the algorithm with enough labeled examples to decide how the features should be used to classify unlabelled examples. The system presented in Chapter 3 is of the semantic type, however certain of the insights gleaned from the successful application of machine-learning models are directly applicable to semantic models (and vice-versa).

This chapter is structured as follows: the first section lists a few of the existing and potential applications of automated sentiment analysis, including commonly used corpora; the second section is concerned with semantic model research, particularly the construction of SO lexicons; the third section focuses on machine-learning and the kinds of features that have been used to build successful sentiment classification models; and the fourth and last section compares the two models in terms of achieved performance, flexibility, and potential.

2.1 Applying Text Sentiment Analysis

The internet has made it possible for individuals outside the professional media to express their opinions on any topic that excites their interest or their ire. At the same time there are other sections of society whose livelihood depends on knowing what people think of them or their product. It is ostensibly of benefit to everyone that the society's producers know what society's consumers are thinking (and people are generally curious about such things), however the sheer amount of information available on the web is daunting. The success of websites like www.rottentomatoes.com, which takes the entire critical response to a movie and boils it down to a percentage, highlights our human desire to know what others are thinking without having to wade through a million web pages to do so.

As it happens, movie reviews have played a central role in automated sentiment analysis thus far: for example, Tong (2001) developed a method for tracking on-line discussion of movies, one of the earliest examples of a text sentiment analyzer. At present, the most widely used corpus for text sentiment analysis is probably the Polarity dataset, a collection of 2000 movie reviews taken from an internet newsgroup (Pang et al., 2002) and balanced for polarity. These reviews have been used by Dave et al. (2003), Mullen and Collier (2004), and Whitelaw et al. (2005), among others. Beside the fact that movie reviews are probably on average more interesting to study than, say, lawnmower reviews, one reason movie reviews have gotten such attention is that they are apparently more difficult to classify automatically (Turney, 2002). This seems to be primarily the result of semantic noise caused by plot and character summaries as well as opinions directed at particular aspects of the movie which are not relevant to the final judgment; an excellent movie might be filled with dark characters and have a devastating ending, while an otherwise terrible movie has a great-looking cast and a gorgeous setting.

The original work of Turney (2002) was not limited to movie reviews; reviews of cars, banks, and travel destinations were also included in his analysis. Taboada and Grieve (2004) used a corpus taken from the same website (www.epinions .com) which also contains reviews of music, books, phones, cookware, and hotels; these texts also form the primary corpus of the present research. Another epinions corpus that will figure in the discussion, a large collection of camera, printer, and stroller reviews, has also been used elsewhere (Bloom et al., 2007). Intuitively, a robust text sentiment analyzer should be able to deal with a variety of domains; however, it is also clear that there is domain-specific information that cannot necessarily be handled by a general system. For instance, as part of a project to track the literary reputations of some famous and less-than-famous authors, Taboada et al. (2008) apply an earlier version of the SO Calculator (introduced in Chapter 3) to early 20th century book reviews, with less than encouraging results; it should perhaps come as no great surprise that long-dead literary reviewers express their opinions in language that is significantly different from that of modern bloggers. It seems that the breadth of sentiment analysis is not only a source of great potential but also one of its most difficult challenges.

Moving beyond commercial product reviews, Spertus (1997) suggests that sentiment analysis could lead to software that would automatically block "flaming" on internet newsgroups; it is worth pointing out that in order to realize applications of this type, more emphasis would need to be placed on classifying the strength of orientation. Turney (2002) notes that a similar system could be used to detect academic peer reviews that use highly polar language, suggesting emotional bias. Hearst (1992) proposes that information related to semantic orientation could supplement the hit count traditionally provided by internet web browsers. Das and Chen (2001) look at using opinions on investor bulletin boards to predict stock prices, while Lerman et al. (2008) similarly predict the political fortunes of candidates. Cho and Lee (2006) have a particularly novel use for sentiment information; instead of focusing on the evaluative aspect, they use affect information from song lyrics in a Korean search engine that allows users to find music appropriate to their mood.

The global potential of sentiment analysis is demonstrated in work like Bautin et al. (2008). They extract named entities (e.g., George W. Bush) and surrounding sentiment from newspapers in multiple languages around the word in order to create a sentiment map that shows how opinion towards an individual or country varies over space and time. This kind of cross-linguistic research will be discussed in more detail in Chapter 5; the remainder of this chapter is focused exclusively on sentiment analysis in English, the language which has received the most attention so far.

2.2 The Semantic Model: Building a Better Dictionary

In the semantic model, the semantic orientation (SO) of a text document is ultimately the sum of its parts. But which parts? Many researchers have focused on individual words, and in particular adjectives, which have shown to be good indicators of subjectivity (Bruce and Wiebe, 2000, Hatzivassiloglou and Wiebe, 2000). Most of the articles discussed in this section are focused on deriving the word SO (or an SO equivalent) automatically from existing linguistic data; unfortunately, relatively little attention has been paid to how well these derived SO values perform in text classification tasks.

One exception is Turney (2002), which not only attempts to classify full texts, but eschews a *unigram* (single word) approach in favor of two-word *bigrams*, extracted according to their part of speech (i.e., adjective/noun pairs, adverb/verb pairs, etc.). The SO values of these bigrams are derived by calculating their Pointwise Mutual Information (PMI), which is defined as follows (Church and Hanks, 1989):

(8) $\text{PMI}(\text{word}_1,\text{word}_2) = \log_2(\text{p}(\text{word}_1 \,\&\, \text{word}_2)/\ \text{p}(\text{word}_1)\ \text{p}(\text{word}_2))$

That is, the PMI of two words is equal to the base-2 log of the probability of the two words appearing together, divided by the product of the independent probabilities of the words; as such, the PMI of two words that appear independently of one another would be close to zero (since $\text{p}(\text{word}_1 \,\&\, \text{word}_2) = \text{p}(\text{word}_1)\ \text{p}(\text{word}_2)$). In order to calculate the SO value of a phrase, Turney uses the PMI of the phrase and two seed words of opposing polarity (“excellent”, and “poor”), with internet hit counts (using the AltaVista search engine and its NEAR operator, which searches for collections of words in close proximity) standing in for the probabilities in (1):

(9) SO(*phrase*) = PMI (*phrase*, “excellent”) – PMI (phrase, “poor”)

Essentially, if a word tends to appear more often with the word “excellent” than the word “poor”, it is probably positive and will, according to the above formula, have a positive SO. Once the SO for each of the extracted phrases in the text has been calculated using the results of internet queries, the average document SO can be calculated.

Turney and Littman (2003) use a slightly modified form of this same algorithm to calculate the SO of individual words, calling the general approach semantic orientation by association (SO-A). They expand their set of seed words to include seven of each polarity, chosen for their insensitivity to context. They also test another measurement of relatedness, Latent Semantic Analysis (LSA), which involves the construction of a matrix representing word occurrence (Landauer and Dumais, 1997); unlike PMI, LSA is able to encode not only the fact that two words appear together, but also whether two words tend to appear near the same words. In general, LSA outperformed PMI, however LSA does not apparently scale up as easily as PMI, and so Turney and Littman were not able to test it on larger corpora (such as all the webpages indexed by AltaVista). The polarity of resulting SO values were compared to the polarity of words from the General Inquirer (GI) lexicon (Stone et al., 1966) and a list of labeled adjectives created by Hatzivassiloglou and McKeown (1997). Identification of word polarity increased above 95% using AltaVista web hits if the test set was limited to the “most confident,” i.e., those words with the most extreme SO values, but less extreme words were more difficult to classify accurately. Randomly choosing seed words from available positive and negative words in the GI lexicon resulted in a sharp drop in performance.

One serious problem with using the internet hit counts to calculate SO is that the internet is constantly in flux. Taboada et al. (2006) report that the AltaVista NEAR operator is no longer available and that the Google search engine (with its text-wide AND operator) is not reliable for the task of calculating SO-PMI; the SO values of adjectives calculated using hit counts from the Google API varied widely from day to day. Kilgarriff (2007) raises a number of concerns about this emerging science of “Googleology,” i.e., trying to get linguistic information from commercial search engines, which are, he points out, simplistic, fickle, and perhaps biased in the information that they provide.

The first major attempt to classify words automatically according to their polarity was probably Hatzivassiloglou and McKeown (1997). Instead of the internet, they used the Wall Street Journal corpus, and only concerned themselves with whether a word was positive or negative. As we saw with (Osgood and Richards (1973), the choice of connectives (i.e., *and*, *but*) joining an adjective tends to indicate whether the two adjectives are of the same or opposing orientation (there is an exception to this, though, namely the conjoining of antonyms, e.g., *right and wrong*). Using counts of adjective conjunctions, the authors derived a dissimilarity value for each pair of adjectives, and then used that to cluster the adjectives into two groups; it has been established that positive adjectives are more frequent, so the cluster whose members were more frequent was chosen as positive. Accuracy was fairly high (92%), though it is perhaps worth mentioning that they deleted any neutral or ambiguous adjectives at the first step, a non-trivial simplification not available to fully automated system.

Several researchers have made use of the lexical database WordNet (Miller, 1990), which groups English words into synonyms sets (synsets), provides a basic gloss for each word sense, and links synsets according to other semantic relations such as hyponomy, holonomy, etc. Kamps et .al (2004) use path length distance in WordNet to derive semantic differential values (Osgood et al., 1957). Basically, they counted the minimum number of synonym relation links intervening between a word and the prototypical examples of each of the three factors (i.e., *good*/*bad* for Evaluation, *strong*/*weak* for Potency, and *active*/*passive*), taking the difference between path length to each of the poles as the corresponding value. The method is simple but relatively effective. Unlike Hatzivassiloglou and McKeown (1997), Kamps et al. took neutral words into account; they found that their accuracy increased significantly (to over 75%) when they treated a wide range of scores around zero as being neutral.

Esuli and Sebastiani (2005) use machine learning techniques to classify individual words as positive or negative using their WordNet glosses. The first step is to derive a set of features (positive and negative words) with enough coverage to train a classifier. This is accomplished using two small sets of seed words (e.g., *good, nice, etc.* and *bad, mean, etc.*, from (Turney and Littman, 2002)) that are expanded iteratively using the WordNet synonym, antonym, hyponym, and hypernym relations. When the set of terms was sufficiently large, the glosses and sample sentences were used to train the classifier. The hypernym relation proved too general, and the hyponym relation was only somewhat helpful; the best results were achieved when the synonyms and antonyms of adjectives alone were used to expand the term sets. Having separate features for negated items (e.g., *not good*) also improved accuracy as compared to the GI lexicon. This research lead to the creation of SentiWordNet (Esuli and Sebastiani, 2006), which makes use of an improved version of the algorithm to provide positive, negative, and objective ratings to all the synsets in WordNet. One obvious problem that SentiWordNet has inherited from WordNet is how fine-grained its synsets are: the adjective “good” has 24 senses, some of which are neutral, and even one which is negative! A great deal of word-sense disambiguation would be necessary to make full use of the information here, and some information about the frequency of occurrence of each sense (in particular domains) would be extremely helpful. Also, it is not clear how well the positive or negative rating of a word corresponds to the strength of positive or negative sentiment; for example, most senses of “good” are actually at least as positive or even more positive than the one sense of “excellent,” whose positive rating is only 0.625 (out of 1).

The Appraisal theory of Martin and White (2005) has also been applied to the semantic model, albeit in a preliminary way. Taboada and Grieve (2004) supplement the SO value of words with information about whether the words seem to be used as Affect, Appreciation, or Judgment; instead of an all or nothing classification, each word has three values that sum to one. They derive these numbers based on how the words are used, assuming that adjectives appearing after first-person copulas (*I was)* are being used to express Affect, adjectives after human third-person copulas (*he was*) are Judgment, and adjective appearing with non-human third-person copulas (*it is*) are Appreciation. Read et al. (2007) are critical of their method, and it is easy to construct examples where this simply does not work (*he was afraid* is affect, *I am intelligent* is judgment), however the method is good enough to identify clear examples of affect (afraid, happy), judgment (weak), and appreciation (great). Unfortunately, there is no evidence presented that the use of Appraisal theory helps the sentiment classification task.

2.3 The Machine Learning Model: Finding the Right Features

Advancements in computer science have lead to the development of a number of machine learning algorithms that can be applied to text classification; the most commonly used include Decision Trees, Naïve Bayes, Maximum Entropy, and Support Vector Machines (SVMs). A detailed discussion of how these algorithms function is beyond the scope of this work, see, for instance, Witten and Frank (2005). The underlying theory, however, is a fairly straightforward. First, a collection of features is extracted from already labeled examples. These features can be numerical or Boolean, for instance how many times the word *good* appears in the text, or simply whether or not it appears. If a particular feature tends to be high (or true) consistently when a text is of a certain type, the algorithm will "learn" that this feature is a good indicator of that type; in the algorithm we will be concerned with, this corresponds roughly to placing a certain weight on that feature. When presented with new (testing) data, the classifier will derive values for the features based on the text, multiply those values by the weights that were learned during training, and sum them together; the numerical result will determine the classification. For example, suppose that during training *good* appeared often in known positive texts, whereas *bad* appeared often in known negative texts, resulting in a +1 weight on *good* and a -1 weight on *bad* after training (let us assume that there were just those two features). When a new text is fed to the classifier, it will count the appearances of *good* (say, 3) and *bad* (say, 5), and decide based on their weighed sum (-2) that the text is negative. In this simple case, the calculation to determine the classification is not really any different than Turney's word-counting model; the differences become more obvious when one widens the scope of features or begins to consider the effect of context.

It is not always obvious which particular algorithm will lead to the best classifier, so, in their initial application of machine learning to the problem of classifying text according to sentiment, Pang et al. (2002) tested Bayes Naïve Classifiers, Maximum Entropy, and Support Vector Machines to see which would best classify the movie reviews in an earlier 1400 text version of the Polarity Dataset. The answer was fairly conclusive: SVMs outperformed the other two algorithms with most combinations of features, and had the highest scores overall. Based on this result, most of the sentiment analysis research based on machine learning has made use of SVMs, and so, except when otherwise noted, the discussion of machine learning that follows assumes a SVM classifier.

Besides looking for the best algorithm, Pang et al. (2002) also tested a number of feature types, including (one-word) unigrams and (two-word) bigrams, with or without appended part of speech tags or indicators of their position in the text. Rather surprisingly, the optimal SVM classifier did best with only unigram features; more complex features generally led to a moderate decrease in performance. Another interesting result was a large performance boost when the unigram features were sensitive only to the occurrence of a word in a text and not its frequency. A dramatic drop in performance was observed when only the 2633 adjectives in the texts were taken as features, much worse than if the total number of unigrams are limited to that same number (i.e., the 2633 most commonly occurring words are chosen as features); more rigorous testing of POS speech filtering by Salvetti et al. (2006) has confirmed that machine classifiers perform best when all (commonly occurring) unigrams are included. Accounting for negation by labeling words that appeared after a negation word with a special tag had a slightly helpful but mostly negligible effect on performance. Otherwise, the best results were achieved using the 16164 unigrams that appeared at least four times in the corpus; the authors note that some of the features that were the most indicative of positive or negative sentiment (e.g., "still" for positive, or "?" for negative) would not have traditionally been viewed as carrying sentiment.

Dave et al. (2003) also experimented with a number of linguistic features in an attempt to improve the performance of a text sentiment classifier; the use of WordNet synsets, negation (along the same lines as Pang et al. 2002), and collocation information all proved ineffective, however word stemming did lead to some improvement. Unlike Pang et al. 2002, the Dave et al. 2003 classifier did much better when bigram and trigram features were used instead of unigrams (Ng et al. 2006 also reports improved performance with bigrams and trigrams). A more flexible approach, which allows the length of the n-gram to increase without bound provided there is sufficient information gain, also showed promise. Other features, such as group of words which appear within *n* words of each other, or n-grams with intervening wildcards, did not show significant improvement over trigrams.

Mullen and Collier (2004) supplemented lemma (i.e., base forms, without inflection) features with two other semantic measures: semantic orientation (SO), using the pointwise mutual information derived from search engine hits (Turney, 2002) and Osgood semantic differential (SD) values (Osgood et al., 1957), i.e., evaluation, potency, and activity, determined using WordNet minimum path lengths (following Kamps and Marx 2002). From each of these, a family of features was created; the text-wide word counting values and the average values of words in various relationships to the topic (either the work or artist in question) were included. When added together, Turney SO values provied consistent improvement while Osgood SD values had little or no effect on performance, however a hybrid SVM which used the output of SVMs trained independently on each of the different sets of features did better than any single SVM alone.

Using the Appraisal Theory of Martin and White (2005), Whitelaw et al. (2005) used features that not only took into account the Orientation (positive or negative) of adjectives in the text, but also their Attitude Type (appraisal, judgment, or affect)and Force (low, neutral, or high). They tested a number of combinations, and got the best results (better than all preceding studies) from a SVM trained on a bag of words plus a set of features that reflected the frequency of "appraisal groups" (adjectives and their modifiers) grouped according to their Attitude Type and Orientation. Not surprisingly, appreciation was the Attitude Type most relevant for predicting sentiment in the movie review corpus. The inclusion of Force features, however, degraded performance.

Ng et al. (2006) also saw significant improvement when they modified *n-gram* features using the orientation of adjectives appearing in the text. After building a lexicon manually from the Polarity Dataset, they added a new set of features that substituted polarity information (i.e., positive or negative) in the place of adjectives appearing in bigrams, trigrams, and dependency relations. They suggest that this method is an effective way of sidestepping the data sparseness problem for bigrams and trigrams (i.e., when there are not enough occurrences of a feature to properly judge whether it is a good indicator for classification).

The work of Riloff et al. (2006) focuses on the notion of feature subsumption and the use of Information Extraction (IE) patterns. One feature is said to subsume another when the set of text spans which matches the first pattern (or string) is a superset of the text spans that match the second. For instance, the feature *good* would subsume the bigram feature *very good* or the IE pattern <subject> *is good*. This relation allows for a subsumption hierarchy where complex features are, by default, subsumed by simpler ones, cutting down on the total number of features. However, if a more complex feature has significant information gain as compared to a simpler one that subsumes it, it is not behaviorally subsumed, and is therefore included in the feature set. A good example is the bigram *nothing short (of)*, which has a positive usage not directly derivable from its unigram components *nothing* and *short*. A modest performance gain was observed in several different corpora using this method in lieu of or together with traditional feature selection (e.g., limiting the total number of features based on frequency of occurrence).

Abbasi et al. (2008) began with a fairly wide range of syntactic (e.g., N-grams, POS) and stylistic features (e.g., appearance of function words, vocabulary richness, even appearance of individual letters), and then showed how a feature selection algorithm based on maximum entropy can be effective in significantly boosting performance above the baseline with all features are included. Their performance in the Polarity Dataset using feature selection (around 95%) is the best reported to date.

2.4 Comparing the Models

Turney (2002) reported an average 74.4% accuracy using his SO-PMI semantic model, with only 65.8% accuracy in the domain of movie reviews. By contrast, the best machine-learning model of Pang et al. (2002) reached 82.9% in an early 1400 text version of the Polarity dataset; others (Whitelaw et al. 2005; Abbasi et al. 2008) have reported results on the 2000 text version above 90%. These results are not directly comparable, however, since different corpora were used. In this section, I discuss research involving direct comparison of the performance of the two models, and round out the chapter with a general discussion of their strengths and weaknesses, focusing particularly on domain flexibility and use of contextual features.

Chaovalit and Zhou (2005) carried out the first side by side comparison of the two approaches, using a small dataset from a movie review website (www.moviejustice.com). For their semantic model, they adopted the bigram PMI approach from Turney (2002), taking account of the negative bias seemingly inherent in the PMI calculation (also noted in Turney) by shifting the cutoff point between positive and negative reviews. Their machine-learning model included unigrams, bigrams and trigrams, minus a stop list of commonly occurring words. They report 77% accuracy using the semantic model on a test set of 100 words, and 85.5% accuracy using the machine-learning model with 3-fold cross validation across their entire corpus of 332 texts; unfortunately, there are no results reported that compare performance on exactly the same set of data. A deeper problem is that they did not make any attempt to control for the number of positive and negative reviews: there were 285 positive reviews, and 47 negative reviews. Although this may reflect the real-world situation, it means that the baseline accuracy (if the algorithm just picked the most likely polarity) is 85.8%, higher than the performance of either model. And, indeed, both models did quite poorly on negative reviews, despite high overall accuracy.

Kennedy and Inkpen (2006) used the entire Polarity dataset (2000 reviews) for both semantic and machine learning testing. They tested an number of combinations of options, finding that the use of contextual valence shifters (Polanyi and Zaenen, 2006) boosted the performance of both models (particularly the semantic model), and that, while the semantic model was very sensitive to the dictionary chosen (adding Google PMI dictionaries decreased performance, for instance), the SVM classifier always did best with lemma unigrams and bigrams; limiting unigrams to the ones in previously existing polarity dictionaries (e.g., the GI) was counterproductive. Overall, the SVM classifier outperformed the term-counting (semantic) method by a large margin: the best term-counting model had an accuracy of only 67.8%, as compared to 85.9% for the SVM classifier. A hybrid SVM classifier trained on the output from each model (comparable to Mullen and Collier 2004) did the best of all, reaching 86.2% accuracy. The authors note that this last performance increase was possible in part because the classifiers seems to make different mistakes; the term-counting model is far better at classifying positive reviews correctly, while the SVM classifier does better on average with negative reviews.

A couple of studies have also been done comparing the performance of the two models in Chinese. The results are contradictory, however: whereas Ye et al. (2005) found that the SVM classifier outperformed a semantic model by about 5% (using the same methodology as Turney and Peng et al.), the performance of the SVM classifier in Ku et al. (2006) was close to chance, leading the authors to reject machine text-classification as a feasible means to carry out sentiment analysis.

Otherwise, there is general consensus in the literature that machine-learning models perform significantly better than semantic models, leading some researchers to reject the latter as a viable approach (Bartlett and Albright, 2008, Boiy et al., 2007). However, it is important to note that although SVM classifiers do very well in the domain they are trained on, performance can drop precipitously (almost to chance) when the same sentiment classifier is used in a different domain (Aue and Gamon, 2005). To show why this might be the case, I extracted the 100 most positive and negative unigram features from an SVM classifier that reached 85.1% accuracy on the Polarity Dataset. Many of these features are quite predictable: *worst*, *waste*, *unfortunately*, and *mess* are among the most negative, whereas *memorable*, *wonderful*, *laughs*, and *enjoyed* are all highly positive. Others are domain specific and somewhat inexplicable, e.g., if the *writers*, *director*, *plot* or *script* is mentioned, the movie is likely to be bad, whereas the mention of *performances*, *ending* and even *flaws* indicates a good movie. Closed-class function words appear frequently: for instance, *as*, *yet, with*, and *both* are all extremely positive, whereas *since, have*, *though*, and *those* have negative weight; names also figure prominently, a problem noted by multiple researchers (Finn and Kushmerick 2003; Kennedy and Inkpen 2006). Perhaps most telling is the inclusion of unigrams like *2*, *video*, *tv,* and *series* in the list of negative words; the polarity of these words actually makes quite a bit of sense, in context: sequels and movies adapted from video games and TV series *do* tend to be worse than the average movie. However, these kind of real-world facts are not really the sort of thing we want a text sentiment classifier to be learning; within the domain of movie reviews it is prejudicial, and in other domains (say, video games or tv shows) it would be a major source of noise. By comparison, one of the goals of the semantic model is to identify words that carry polarity mostly independent of context; by relying on hit counts, SO-PMI, for instance, can represent a much broader range of usage than is possible in a machine-learning model which is trained on domain corpus data. Following on this idea, Andreevskaia and Bergler (2008) are able to improve cross-domain performance when they supplement a domain-dependent classifier with a more general lexical system built using WordNet glosses (Andreevskaia and Bergler, 2006).

Another area where the semantic model might be preferable to a pure machine-learning model is in simulating the effect of linguistic context. Valence shifters, for instance, are probably less useful to an SVM classifier because they require an increase in the number of features, with each feature requiring further independent examples. SVM classifiers (e.g., Kennedy and Inkpen 2006) generally deal with negation and intensification by creating separate features, i.e., the appearance of *good* might either be *good* (no modification) *not_good* (negated *good*) *int_good* (intensified *good*) or *dim_good* (diminished good). However, the classifier cannot know that these four "types" of *good* are in any way related, and so in order to train accurately there must be enough examples of all four in the training corpus; for *good* this might in fact be the case, but for other less common adjectives this would be a serious problem, one that would only get worse if more context was considered (e.g., stronger intensifiers like *extraordinarily* versus weaker intensifiers like *very*). As we will see in Chapter 3, semantic models can deal with negation and intensification in a way that generalizes to any word in their dictionary, with a corresponding increase in performance.

There is other, higher-level contextual information that is likely to play a key role in improving the performance of sentiment analyzers; in particular, the use of other types of text classification (which will be our focus in Chapter 4). While we saw that the addition of position features degraded the performance of an SVM classifier (Pang et al. 2002), semantic models respond favorably to the use of weights to boost the SO value of words appearing in parts of the text likely to contain opinions (Taboada and Grieve 2004). In general, while deleting potentially relevant input or creating overly complex (difficult to train) features is the only obvious way to use this kind of context in the standard machine learning model, a semantic model can integrate this information in a less disruptive fashion using weights on SO values of words appearing within the scope of relevant text spans. It is also worth noting that the use of discourse information in semantic models might result in the sort of improvement that SVM models derive from taking closed-class function words into account.

Finally, there is the third sub-task of sentiment analysis: determining the strength of orientation. SVM classifiers are not naturally suited to determining both the direction and strength of orientation, since their classification is binary; trying to identify strength independent of direction using machine learning has shown only modest results (Wilson et al., 2004), and capturing the nature of rating scales using SVMs, while not impossible, does not come naturally to the model (Pang and Lee, 2005). On the other hand, since the semantic models output a numeric SO value that averages individual SOs over the whole text, they should be able to capture a continuous spectrum of force, provided the SO values of individual words properly reflect the strength and well as orientation of sentiment. We will explore this question later in Chapter 3.

Despite the potential advantages of the semantic model, there is a significant performance gap to be overcome if it is to become a feasible method for further research. In the next chapter, I will describe the creation of a semantic system that closes some of that gap, showing good cross-domain performance and enough flexibility that additional progress is to be expected with the addition of external modules.

Chapter 3: The Semantic Orientation Calculator

In this chapter, I describe the semantic orientation calculator, or SO Calc, which uses low-level semantic and syntactic information to calculate the overall polarity of product reviews. As part of this thesis I completely re-wrote SO Calc in the Python programming language and added most of the features discussed here, however this was an adaption of the previous Perl version that implemented the basic word-counting algorithm, text position weighting, and a simpler form of negation[2]. In Section 1, I discuss the development of SO Calc and its various features, including the creation of dictionaries and handling of negation, intensification, modality, repetition, and other phenomenon that may affect semantic orientation. Section 2 presents the results of testing on three different corpora, primarily to show how various options improve performance, but also looking at other aspects, such as performance in terms of ranking scales.

3.1 Calculating Sentiment

Following Osgood et al. 1957, the calculation of semantic orientation begins with two fairly major assumptions: that individual words have what is referred to as *prior polarity*, e.g., a semantic orientation that is independent of context, and that said semantic orientation can be expressed as a numerical value. As we saw in the previous chapter, several lexicon-based approaches have already adopted these assumptions in one form or another, and in what follows the advantages of this explicit quantification will be made clear. The SO values contained in SO dictionaries serve as the basis for a more complicated system that involves modification of these values based on contextual information, which will be the main focus of this section.

3.1.1 Open-Class Dictionaries

The development of SO Calc (or SO-Cal) prior to my involvement was primarily focused on the automated creation of adjective dictionaries (Taboada e al. 2006), which we discussed in Section 2 of Chapter 2, and various uses of discourse information (Taboada and Grieve 2004; Voll and Taboada, 2007), a subject which we will return to in Chapter 4. The current version uses manually-tagged dictionaries, where a rater (one of the researchers[3]) assigns each word or phrase an integer SO value between 5 and negative 5. In general, words with zero SO are simply omitted from the dictionary. Table 1 includes a sample adjective for each rating, and a SO value derived for each word using Turney's SO-PMI method (cf. Taboada et al. 2006).

[2] Jack Grieve, Katia Dilkina, Caroline Anthony, and Kimberly Voll all contributed to this earlier version.

[3] Much of the Epinions-derived adjective dictionary was originally rated by Kimberly Voll; I have been the main rater for all the words added since September 2007, including several hundred adjectives and all nouns and verbs.

Word	Manual SO	Google SO-PMI
brilliant	5	0.62
engrossing	4	-0.13
amusing	3	-1.52
consistent	2	2.82
functional	1	2.95
genealogical	-	5.14
tame	-1	-0.07
corny	-2	-2.08
stagnant	-3	-4.13
obscene	-4	-0.60
horrendous	-5	-3.52

Table 1: The SO Scale

That the top SO-PMI score in the table is *genealogical* highlights one of the major problems with an unsupervised approach to dictionary building that relies on word context; many words having no particular evaluative content will be assigned (often extreme) SO values, which leads to a great deal of noise in the final calculation. As for the words which are evaluative, the polarity of the derived SO is correct more often than not, but they often do not properly reflect the strength of evaluation. The use of manual- ranking mostly overcomes these problems, since the ranker knows intuitively which words carry positive and negative sentiment, and can compare words to each other to determine where they should fall on a scale. One quick way to determine relative SO is to construct sentences like:

(10) a. It's not just amusing, it's brilliant (amusing < brilliant)
b. ?? It's not just horrendous, it's poor (poor < horrendous)

For words whose primarily loading is on the evaluative dimension (to use Osgood's terminology), these sentences seem good only when the two words are of the same polarity but the second word is stronger than the first. However, the test can also fail when words conflict with respect to some other semantic dimension.

(11) a. He's not just mean, he's evil (mean < evil, no conflict)
b. ? He's not just timid, he's evil (timid and evil are opposites in terms of potency)

The use of a 10-point scale (excluding zero) for manual ranking reflects a tradeoff between, on the one hand, a desire to capture clear differences in word meaning, and, on the other, the difficulty in assigning extremely fine-grained values to out-of-context words. As the granularity increases, the choices become more arbitrary, more subject to individual variation. For example, consider trying to give strict evaluative rankings to the words *awesome*, *delightful*, *fantastic*, *magnificent*, *fabulous, and sensational* (all 5s in our dictionary); it might be possible, but it is significantly less obvious than *phenomenal*, *good*, *fair*, *poor*, *bad*, and *awful*, and making such minute distinctions is in any case unlikely to be of much benefit from the perspective of sentiment analysis.

Though for simplicity we have so far looked only at adjectives, all open-class words (including nouns, verb, adjectives, and adverbs) have strongly evaluative exemplars, and so we have augmented the SO Calculator with dictionaries for each part of speech (POS); texts are tagged with a rule-based tagger (Brill, 1992) prior to processing. Segregating words by POS has the additional benefit of providing very basic word-sense disambiguation; for example *good* appears in the adjective and adverb dictionaries but not in the noun dictionary, eliminating *goods* (meaning *merchandise*) from the calculation of SO. For the most part, the scores for adverbs were directly derived from their adjective counterpart, i.e., *poor* (-2) becomes *poorly* (-2), except in cases where there is a clear difference in evaluative content of the two forms (*e.g., ideal*, *ideally*). Some examples are given in Table 2:

Word	SO
excruciatingly	-5
inexcusably	-3
foolishly	-2
satisfactorily	1
purposefully	2
hilariously	4

Table 2: Sample Adverbs

Nouns and verbs are often morphologically related to adjectives (consider *lost*, *lose*, *loser*), but in this case their SO values were assigned separately, though effort was made to keep SO values consistent across POS, when appropriate. SO Calc has a built-in lemmatizer for the regular inflection of nouns and verbs, so only the base form of most words needs to be included in the dictionary. Some examples are given in Table 3:

Word	SO
hate(verb)	4
hate (noun)	4
inspire	2
inspiration	2
masterpiece	4
fabricate	-2
sham	-3
relish (verb)	4
determination	1

Table 3: Sample Nouns and Verbs

Note that the relatively low rankings for *inspire* and *inspiration* reflects an averaging across possible interpretations:

(12) a. This film inspired me to pursue my dreams.
b. This film was inspired by true events.
c. I don't know what inspired them to make this film.

(12a) is probably the most common usage (and, independently, would probably be ranked higher than a 2), however, not wanting to integrate a full system for word sense disambiguation at this stage, the best approach seemed to be to split the difference in those cases when more than one word sense came immediately to mind. The rankings for nouns and verbs were, on average, lower than adjectives and adverbs, reflecting the fact their use is often not explicitly evaluative to the same degree; many have both polar and neutral readings.

Finally, I added support for multiword expressions, which were integrated directly into the existing dictionaries based on the POS of the head word. Multi-word expressions automatically take precedence over single-word expressions; for instance *funny* by itself is assumed to be positive (+2), but if the phrase *act funny* appears, it is given a negative value (-1). The expressions can include wildcards, multiple options, or tags rather than words, allowing us to capture most syntactic variations (for instance, the possibility of a short intervening noun phrase within a phrasal verb). In the version of SO Calc described here, the number of multiword expressions in open-class dictionaries is relatively small (less than 200), reflecting only a preliminary investment of resources ; the majority are particle verbs like *wear out*, *pass off*, *blow away*, and *spice up*, collected from in a phrasal verb dictionary. Ideally, these could be discovered automatically using N-grams or extraction patterns, and assigned a special SO value only in those cases where the SO value of the whole is clearly not equal to the sum of the parts.

The current version of SO Calc has four open-class dictionaries and one closed class-dictionary of intensifiers (the make-up of which will discussed later in the chapter); the size of the dictionaries are given in Table 4**.**

Dictionary	No. of Entries
Adjectives	2257
Adverbs	745
Nouns	1142
Verbs	903
Intensifiers	177

Table 4: The Size of SO Calc Dictionaries

The words in the dictionaries were drawn from a variety of sources, but for adjectives the primary source was the 400 epinions reviews (Epinions) from Taboada and Grieve 2004—all the words that were tagged as adjectives by our tagger were extracted and given SO values, when appropriate. When adding nouns and verbs, we drew also from the General Inquirer (GI) dictionary and from a small subset of the Polarity Dataset (Movies). This gave us a fairly good range in terms of register: the Epinions and Movie reviews were written casually by members of the general public, containing lots of informal words like *ass-kicking*, *unlistenable* and *nifty*; on the other end of the spectrum, the GI was clearly built from much more formal texts, and contributed words like *adroit* and *jubilant*, which we wouldn't expect to see often in product review corpora, but might be more useful in tasks such as literary sentiment detection.

It is difficult to directly measure the coverage of our dictionaries, since there is no direct way to estimate the number of SO-carrying words and expressions in English (though clearly it should be significantly larger than 5000). Wilson et al. (2005) provide a list of subjectivity cues with over 8000 entries, however there are many neutral, repeated, and inflectionally-related entries, with many more nouns than we have, and far fewer adjectives. With an earlier form of SO Calc, we did a simple test of coverage: we took 50 texts from the Polarity Dataset (texts which we had not previously drawn words from) and extracted all words judged to have sentiment that were not already in our dictionaries. We found 116 adjectives, 62 nouns, 43 verbs, and 7 adverbs, which amounts to less than 5% of the current dictionary (these words are included in the current version). What is particularly interesting is that the inclusion of those words in our dictionary actually degraded performance in that dataset (by 4%), suggesting that coverage might be a double-edged sword, particularly for opinion texts that involve a lot of pure description, like movie reviews. In any case, the best argument for good coverage is acceptable performance for new texts in new domains, and indeed we will see later (in section 2 of this chapter as well as section 2 of Chapter 5) that there is almost no difference in performance between texts which were used to build our dictionary, and others that were not.

In order to minimize the subjectivity of the rankings, the dictionaries were reviewed by three other researchers[4], and when there was disagreement over the ranking of particular word a consensus was reached.

SO Calc has several options with respect to dictionaries. First, POS dictionaries and multi-word dictionaries can be disabled, so that they are not taken into account in the calculation. SO Calc also allows for use of genre-specific dictionaries that supplement the main dictionaries—in these cases, words in the genre-specific dictionary will override any value in the main dictionary. There is also an option to simplify values, changing the 5 to -5 scale into the more common binary classification schema; this also has the effect of simplifying the intensification modifiers, discussed in the next subsection.

3.1.2 Intensification

Quirk et al. (1985b) classify intensifiers into two major categories, depending on their polarity: *amplifiers* (e.g., *very*) increase the semantic intensity of a neighbouring lexical item whereas as *downtoners* (e.g., *slightly*) decrease it. The contextual valence shifter approach of Polanyi and

[4] Milan Tofiloski, Vita Markman, and Yang Ping

Zaenen (2006) was to model intensification using simple addition and subtraction of fixed values. One problem with this kind of approach is that it does not obviously account for the wide range of intensifiers within the same subcategory. *Extraordinarily*, for instance, is a much stronger amplifier than *rather*. Another concern is that the amplification of already “loud” items should involve a greater overall increase in intensity when compared to more subdued counterparts (compare *truly fantastic* with *truly okay*); in short, intensification should also depend on the item being intensified. In our system, intensification is modeled using modifiers, with each intensifying word having a percentage associated with it; amplifiers are positive, whereas downtoners are negative, as shown in Table 5.

Intensifier	**Modifier %**
Slightly	-50%
somewhat	-30%
pretty	-10%
Really	+15%
Very	+25%
extraordinarily	+50%
(the) most	+100%

Table 5: SO Modifiers for Certain Intensifiers

For example, if *sleazy* has an SO value of -3, *somewhat sleazy* would have an SO value of

-3 + (-3 X -30%) = -2.

If *excellent* has a SO value of 5, *the most excellent movie I’ve seen this year* would have an SO value of

5 + (5 X 100%) = 10

Intensifiers are additive. If *good* has a value of 3, then *really very good* has an SO value of

3 + (3 X 15%) + (3 X 25%) = 4.3

As with the SO dictionary values, there is a fair bit of subjectivity associated with assignment of a modifier value. Again, the *it’s not just* test can be helpful:

(13) a. It’s not just *very good*, it’s *extraordinarily good*.
b. ? It’s not just *very* good, it’s *really good*.
c. ?? It’s not just *very good*, it’s *pretty good*.

Since our intensifiers are implemented using a percentage scale, they can fully reflect the variety of intensifying words as well as the SO value of the item being modified. This scale can be applied to other parts of speech, given that adjectives, adverbs, and verbs can all use the same set of adverbial intensifiers:

(14) a. The performances were all really fantastic.
b. Zion And Planet Asai from the Cali Agents flow really well over this.
c. I really enjoyed most of this film.

Some verb intensifiers appear at the end of the clause, a fact that we integrated into our model—for verbs, we search both directly before verb as well as at the nearest clause boundary after the verb.

(15) a. The movie completely fails to entertain.
b. The movie fails completely.

Nouns require special attention, as they are modified by adjectives and quantifiers, not adverbs. We are able to take into account some kinds of adjectival modification using our main adjective dictionary (e.g., *nasty problem = nasty + problem*); however there are a small class of adjectives which would not necessarily amplify or downtone correctly if considered in isolation:

(16) a. The plot had *huge* problems.
b. They have made *major* progress.
c. This is a *total* failure.

Here, adjectives such as *total* do not have a semantic orientation of their own, but like adverbial intensifiers they contribute to the interpretation of the word that follows it; *total failure* is presumably worse than just *failure*. Thus our intensifier dictionary also includes a small set of adjectival modifiers (often the adjectival form of an adverb already in the dictionary); when an intensifying adjective appears next to an SO-valued noun, it is treated as an intensifier rather than as a separate SO-bearing unit. This treatment can be extended to quantifiers in general:

Intensifier	**Modifier %**
a (little) bit of (a)	-50%
a few	-30%
minor	-30%
some	-20%
a lot	+30%
deep	+30%
great	+50%
a ton of	+50%

Table 6: SO Modifiers for Certain Noun Intensifiers

Other words that have a negation-like effect can be captured as intensification. Comparatives like *less*, superlatives like *the least*, and adverbs like *barely*, *hardly*, *almost*, which in general indicate a failure or insufficiency with respect to an attribute, are handled using modifiers of below -100%, resulting in a polarity change:

(17) a. It was less interesting than I was expecting (3 + (3 X -150%)) = -1.5
b. It had the least coherent plot imaginable (2 + (2 X -300%)) = -4
c. I was hardly able to stay awake it (1 + (1 X -150%)) = -0.5

Comparatives in general are troublesome, because they are so context dependent. Our approach is to minimize their effect, always treating them as downplayers (*more* = -50%, *merrier* = 2 + (2 *-50% = 1); in fact, performance improves when *better* has no SO value at all! Superlatives are amplifiers, but are only active in the presence of definite determiners, a restriction which eliminates a few confounding instances (e.g., *best actress winner Julia Roberts*).

SO Calc supports the specification of words that intensify at a clause level. At present, the only word that has been used for this purpose is *but*, which (by default) doubles the SO of words appearing after it but before a clause boundary:

(18) I enjoyed (+3) the villain, but otherwise the acting was mediocre (-2 X 2). 3 -4 = -1

In rhetorical structure theory (Mann and Thompson, 1988), (18) is a concession relation, and the information included after the *but* (the nucleus of the relation) is more central to the author's purpose, and thus can safely be given more weight. In lieu of a full discourse parse, this kind of intensification allows us to capture some simple discourse-level emphasis.

Finally, there are two kinds of intensification that are not signaled by the presence of intensifying words. Exclamation marks are used to indicate increased emotion, and, in informal texts, so do capital letters. (19) is an extended example from our corpus.

(19) We did not book a hotel on South Beach because of the countless "Best Of" shows on the Travel Channel. South Beach always scores in the top 10 beaches in the world and is deemed "best party/best nightlife beach". We did not want drunken college kids walking next to our window all day and night. WERE WE MISTAKEN! We kicked ourselves the entire stay for not staying down there because the beach was 100X more beautiful than the hotel's, there were hundreds of restaurants, and the area had some of the most stunning architecture we had even seen.

This negative text is likely to be mistagged as positive unless *mistaken* is given proper weight. By default, we double the SO value of words contained within exclamation marks and any words that are in all capital letters. In this case, mistaken would have a SO value of -2 *2*2 = -8.

The full list of intensifiers is included in Appendix 1.

3.1.3 Negation

For a certain perspective, negation is a simpler prospect than intensification, being essentially binary in nature. However, the syntax and pragmatics of negation actually make it a much more difficult to model accurately. Consider the examples in (20)

(20) a. Nobody gives a good performance in this movie.
b. Out of every one of the fourteen tracks, none of them approach being weak and are all stellar.
c. Just a V-5 engine, nothing spectacular

First of all, there are a number of words that explicitly signal negation, including *not, no, none, nothing, nobody*, *never*, *neither*, and *nor*. We have added to this list words that also explicitly indicate a negation of possession (of a property), like *without*, verbs *lack*, *miss*, and nouns *lack* and *absence (of)*. However, allowing negation to include a wide variety of negators means that it can be very difficult to detect what is negating what. In (20a), *good* is the SO-carrying item, and is properly negated by *nobody*, since the sentence implies that the performances were not good. However, if *bad* was inserted between *this* and *movie*, the presence of *nobody* would not indicate a negation of that sentiment (more on this later). Similarly, in (20b), we would like *none* to negate *weak*, but not *stellar*. A full syntactic parse might be of some value (though state-of-the-art statistical taggers, trained on newspaper articles, do not do all that well on our informal corpus), but would not really be a substitute for a full *semantic* parse, which unfortunately is not available to us.

An earlier version of the SO calculator (prior to my involvement) just looked back a certain number of words for a negator, a simplistic approach. The latest version includes two options: look backwards until a clause boundary marker is reached (which includes punctuation and all kinds of sentential connectives, including ambiguous ones like *and* and *but*), or look backwards as long as the words/tags are in the backward search skipped list, with a different list for each POS. The former is fairly liberal, and will allow negation at some distance, for instance capturing the effects of verb negation raising (Horn, 1989).

(21) I don't wish to reveal much else about the plot because I don't think it is *worth* mentioning.

Here, a backwards search from *worth*, looking for clause boundaries, would go as far as *because*, finding the *n't* and negating appropriately. The other approach is more conservative; for an adjective like *worth*, the search will go only as far as *it*—for adjectives, copulas, determiners, and certain basic verbs are on the skipped list (allowing negation of adjectives within VPs and NPs, as in (20a), but pronouns are not. Similarly, verbs allow negation on the other side of *to*, and nouns look past adjectives as well as determiners and copula. At present, coordinated negation is not accounted for by either method, except when it is marked with *nor*.

The most straightforward way of representing negation in our quantificational framework is a polarity switch: 3 -> -3. However, in many cases this does not capture the pragmatic interpretation of a phrase. In (20c), for instance, *nothing spectacular* would, under a polarity switch, have a rating of -5, a far cry from the ambivalence that is actually communicated by the phrase. On the other end of the spectrum, *functional* is a 1, but *not functional* seems somewhat worse than a -1. In short, saying something does not meet very minimal positive standard means you are actually making a much more negative statement than if you are saying does not reach a very high positive standard. Horn (1989) captures these subtleties using the notion of *contradictory* versus *contrary* negation; contradictory negation (which is the kind of negation seen in morphological derivations such as *unhappy*) involves a complete polarity flip and does not allow for a middle interpretation, whereas contrary negation allows for a number of interpretations. For example, *unhappy*, like *sad*, is the true polar opposite of *happy* but *not happy* could potentially mean anything from just *content* all the way to *miserable*. Another way to look at it would be to think of (contrary) negation as creating a range containing everything that does not reach the standard; for a +3 word, this would consist of everything from 2 to -5. For weakly positive words, the midpoint of this range (the default interpretation) lies well into negative range, whereas for strongly positive words it would be close to zero. This implies that you would hardly ever get an extremely negative default interpretation for negation, which seems to be the case (you can perhaps imply but not directly express words like *horrendous* using negation); in essence, using a negator allows you to remain basically non-committal[5]. The current version of SO Calc captures this dynamic using an SO shift, shifting the SO value of a word by a fixed amount in a direction of opposite polarity. Some examples from the Epinions corpus are given in (22), using the default shift value of 4 (for adjectives).

(22) a) She's not terrific (5 - 4 = 1) but not terrible (-5 + 4 = -1) either.
b) Cruise is not great (4 - 4 = 0), but I have to admit he's not bad (-3 +4 = 1) either.
c) This CD is not horrid (-5 + 4 = -1).

In the right pragmatic context, (22c) certainly could be construed as demonstrating a positive sentiment, but a more likely interpretation, in my view, is that the CD is still somewhat lacking. On the other end of the spectrum, (23) gives the calculation for the negation-shifted values of some low-intensity words:

(23) a) not average (1 – 4 = -3)
b) not fair (1 -4 = -3)
c) not frequent (1 – 4 = -3)
d) not palatable (1 – 4 = -3)
e) not warm (1 – 4 = -3)
f) not bumpy (-1 + 4 = 3)
g) not heavy (-1 + 4 = 3)
h) not inexperienced (-1 + 4 = 3)
i) not shy (-1 +4 = 3)
j) not rigid (-1 + 4 = 3)

Sometimes shifting seems to work well, particularly for certain commonly negated positive words, but negating a negative word rarely seems to result in a phrase that would be judged by

[5] This is where Potts (2007) theory of narrowing intervals might be valuable; rather than committing to a single (average) SO value, the result of negation (and other contextual effects) could be a fairly wide range of SO values.

humans as being truly positive. Somehow, just by using the negative word, the situation has already been cast in a negative light. Also, negating a negative term to express positivity when there is a positive term available seems a deliberate flouting of the Gricean maxim of manner (Grice, 1975), i.e., be clear in your contribution. This suggests that you are saying more than you are saying; you could be hedging to avoid a bold face lie (*flexible* would be too strong a word, whereas *not rigid* allows for a weaker interpretation), an explicitly negative statement (by *not shy* you really mean *shameless*), or perhaps an immodest one (by *not inexperienced* you really mean *complete mastery*). As we saw in Section 1 of Chapter 1, polarity in human language often defies the pure symmetry that a mathematical approach to a certain extent demands. In any case, in order to reign in the shifting of low-intensity words, SO Calc includes an option to limit negation shifting to no more than the polarity-switch value (so *not rigid* shifts only as far as 1, not 3).

Negation interacts with intensification in important ways, with the two often appearing together in the same expression. To show how SO Calc deals with this, we will work through a fairly complicated example:

(24) There will be a lot of not very merry people this Christmas.

In our dictionary, the SO value of *merry* is 2. *Very* is a +25% modifier, which means that the SO value of *very merry* is 2 + 2*(25%) = 2.5. Next, negation is applied, 2.5 – 4 = - 1.5. Intensifiers highlight the advantages of shift negation: intuitively, *not merry* is more strongly negative than *not very merry*, but a polarity switch would give the opposite effect, 2.5 -> -2.5 (unless you also negative the modifier, turning it into a downplayer). SO Calc searches for intensification after negation as well as before, applying the intensification on the result after negation; in this way, *a lot of* (a 30% intensifier) intensifies *not very merry*, -1.5 + (-1.5*30%) ~= -2. In this case, the two intensifiers essentially cancel each other out. Negation external intensification is fairly restricted (for instance, you never see it with strong intensifiers like *extraordinarily*), but does occur quite often with *really*, e.g., I *really didn't like this movie*, and it is crucial that negation and intensification be applied in the correct order. One further example:

(25) This discussion of negation isn't the least bit interesting.

(25) is a very negative statement (about as negative as negation can get), however at first glance it seems like effects of the negation (*isn't*) and *the least*, which is a negating downplayer, would cancel each other out. In this case, a step by step outward calculation will fail because *the least* modifies *bit*, not *interesting*. In order to make this work, *the least bit* would have to form a single intensifier, a very strong (-99%) downplayer, which is then shifted to become a very strongly negative (~-4). This makes sense, semantically, as *the least bit* seems to be picking out the smallest possible amount of something (like *a little bit,* but more so), and further supports the use of SO shift negation over polarity shift, which would give us an opposite result, with *not interesting* stronger than *not the least bit interesting*. Alternatively, the entire expression *not the least bit* could be included in the dictionary as a negating downplayer (like *the least*) instead

of *the least bit*, which only appears in negated contexts, however this would fail in cases like (26), where negation occurs at a distance.

(26) None of this is the least bit relevant.

3.1.4 Irrealis Blocking

The term *irrealis* is traditionally used to refer various syntactic moods that cause a reader (listener) to interpret a proposition as not being actual. Except for the (essentially) defunct subjective, English lacks inflected moods, however word order, modals, and private-state verbs related to expectation have essentially the same function. A clear example of irrealis in English is the imperative, where it is understood that the action being referred to has not occurred:

(27) Write a good script before you start filming.

Imperatives in English are distinguished, of course, by the appearance of a bare verb at the beginning of the sentence, which is rare in indicative contexts, except in highly informal writing. Taken literally, (27) makes no direct claim about whether a good script was or will be written (though in many contexts it implies otherwise), it merely expresses the strong preference of the author for the proposition to have been true or to become true. The same idea could also potentially be expressed in a number of other ways, depending on how (27) is temporally interpreted and the level of politeness that is desired:

(28) a. I would have liked you to write a good script before you start filming (past).
b. I want you to write a good script before you start filming (future).
c. It would have been nice if you had written a good script before filming (past)
d. It would be nice if you wrote a good script before filming (future).
e. Couldn’t you have written a good script before filming (past)?
f. Can you write a good script before filming (future)?
g. Why didn’t you write a good script before filming (past)?
h. Why don’t you write a good script before filming (future)?

We can see here that modals, verbs that indicate preference, conditionals, and questions are all common markers of *irrealis.* Note that past and future refer to the time when the action would have/will occur, not the time frame of the intention, which is presumably the time of writing (speaking). Even though the past-focused interpretation of (27), made implicit in (28a), (28c), (28e), (28h), does logically imply that the action did not occur, the ostensible purpose of the utterance is not to make such a claim, but instead to express the preference of the author. For word-based sentiment analysis, this is all extremely problematic, because although there is often positive and negative sentiment being expressed in these kinds of sentences, the sentiment is being expressed in a rather roundabout way: the preference communicated by (27) and (28) will sometimes imply disapproval of events that are understood to have not occurred. In some cases the sentiment might be captured by a reversal of the SO value of words appearing in the predicate, but not always; looking only at modals for the moment, it is fairly easy to find examples of both in the movie reviews from the Epinions corpus.

(29) a. I thought this movie *would* be as good as the Grinch, but unfortunately, it wasn't.
b. The film *could* have easily descended into being a farce, but he kept it in line.
c. I am left with an empty feeling where the creepiness *should* be after a good ghost story.
d. This *would* have been welcoming topic to play upon even if it has been done before, but…
e. Well this is a movie that you *would* want to see to bring back that Christmas cheer.
f. But for adults, this movie *could* be one of the best of the holiday season.
g. I know that, even at 11, I *would* have been disturbed by the sexual jokes in it.
h. But you *may* groan as much.
i. My new wife thought it *would* be fun to see so I agreed to see it with her.

The sentiment analysis of (29a-d) would, it turns out, benefit from a reversal of sentiment, whereas for (29e-h) the opposite is true, and for (29i), it is quite clear that *fun* should not be including in the evaluation at all. In general, although there are patterns, identifying the relationship that exists between the SO value of words appearing within an *irrealis*-marked expression and the overall sentiment communicated by the phrase would require a deeper semantic representation than is currently possible with SO Calc. At present, the best approach we have is to block the SO value of these words. Besides modals, we also block the SO value of words appearing after a number of other *irrealis* cues, including conditionals, negative polarity items (NPIs) like *any* and *anything*, certain verbs, questions, and words enclosed in quotes, which are almost always external to the author's opinion.

(30) a. *If* you *want* a nicer Christmas movie, "Elf" is still playing across the hall.
b. So *anything* positive about this movie*?*
c. I remember seeing her and just *expecting* her to be playing an uptight, prim and proper lady.
d. So please disregard all those *'*not historically correct' complaints you may read.
e. I *used to* have great love for Mr. Jon Favereau.

There is good reason to include NPIs as *irrealis* blockers rather than full-fledged negators: NPIs often appear in imbedded alternatives which are not generally marked with question marks and where negation would not be appropriate.

(31) I wonder whether there are going to be any problems with that.

(30e) is not irrealis per se; *used to* unambiguously signals a state of events that was true in the past, but not in the present (i.e., 30e necessarily implies that "I" no longer has great love for Mr. Jon Favereau). Similarly, the use of the perfective *have* with certain modals and verbs such as *should have*, *could have*, *have expected*, *have hoped* also unambiguously signal that the proposition was a preference or expectation that was thwarted. There might be a better way to deal with these than simply blocking the SO value, so we can capture cases like (29b) and (29d) (but also see (29g)). And, for possibility modals like *may*, *might* and *could*, treating them as downplayers rather than blockers would also us to get some of the meaning intended by (29f) and (29h). In short, our preliminary handling of modality leaves some room for improvement.

There is one case, at least, where it is clear that that the SO value of a term should not be nullified by an irrealis blocker:

(32) …can get away with marketing *this amateurish crap* and still stay on the bestseller list?

Though not very common, this kind of off-hand opinion, buried in a question, imperative, or modal clause, is often quite strong and very reliable. SO Calc looks for markers of definiteness within close proximity of SO-carrying words (within the NP), and ignores irrealis blocking if one is found.

One other kind of irrealis blocking included in SO CALC has to do with the case when a SO-carrying term directly modifies (precedes) an SO-carrying term of opposite polarity. We discussed this kind of polarity clash in the context of Osgood in Chapter 1. Our observation is that, when the modifying term is of sufficient strength, it will essentially nullify the evaluative effect of the word it modifies. For example:

(33) a. …read Dr Seuss 's delightfully sparse book (sparse, -1, is blocked)
b. …he proceeds to sit there being as gross as possible (possible, 1, is blocked)
c. A good old-fashioned straightforward plot… (old-fashioned, -1, is blocked)
d. Tom is too pretty for this kind of role (pretty, 2, is blocked)

The word *too* is an interesting case, since it is not entirely clear whether it should be an SO-carrying adverb, a negator, or an intensifier:

(34) a. The ending was too dark for me
b. Actually, it’s not too bad.
c. That man is a little too strong, if you ask me.
d. I liked it too.
e. This is too cool!
f. No, this tea isn’t too strong for me.

Our current approach is to first eliminate the (34d) type of *too* from consideration (it always occurs at the edge of a clause) and treat *too* just like any other negative adverb with a -3 SO value. The major advantage of this is to allow for *too* to contribute in cases where the word it is modifying has no SO value (e.g., *too much*). In (34d), the positive interpretation of *strong* is blocked by *too*. We handle the expression *not too* by having a special negating downplayer (-150%) in our intensifier dictionary, *not too bad* = 1.5. This does not, however, handle outliers like (34e) and (34f), where special interpretation is needed to get the right meaning: in (34e), *too* is a strong amplifier with no negative connotation, and in (34f), *strong* is negative in the context of tea and the limiting *for me*.

3.1.5 Text-level Features

Whereas the improvements discussed in sections 1.1-1.4 would be equally relevant to sentence-level analysis (and I have justified them using short examples from the text), there are other ways to improve text polarity detection besides improving the reliability of individual SO values; this section discusses features which are intended to improve the ability of the SO Calculator to identify polarity at the level of the text. Before addressing each feature in turn, however it is important to understand a little bit more about how SO Calc works. The result outputted by the main calculator is not, in fact, a binary polarity, but rather a text SO value that reflects the average of all the words in the text that had a non-zero SO value after all modifiers were taken into account. That SO value is then compared to a cutoff value (by default, 0) and the text is tagged as positive or negative based on which side of the cutoff it appears.

Appendix 2 contains a full review text and the rich output after analysis by the SO calculator.

3.1.5.1 Cutoffs and Negative Weighting

On average, there are almost twice as many positive words as negative words in our texts (see Table 7 in the next section), which reflects the general human bias towards positive language, noted by Boucher and Osgood (1969). Unless action is taken to counteract this effect, the SO Calculator will always do much better on positive texts than negative texts; in negative texts the positive words often outnumber the negative ones, whereas the reverse is rarely true. SO Calc includes two different ways to counter this bias: one, the cutoff between positive and negative texts can be shifted toward the positive, the approach also taken in Voll and Taboada (1997). For instance, a text calculated to have 0.14 SO would be positive with the default cutoff of 0, but would be negative if the cutoff between positive and negative was shifted to 0.2. One drawback to this approach is that theoretically a text with no negative expressions could be labeled negative simply because it wasn't sufficiently positive (although in practice this is highly unlikely). The alternative, which has become our default approach, involves applying additional weight (by default, +50%) to any negative expression appearing in the text. Intuitively, this is more psychologically plausible, especially given the positive language bias already mentioned; because negative words and expressions are marked, when they do appear they are automatically assigned more cognitive weight. This weight is given at the last step in the calculation, and is applied based on the SO value at that point; for instance, *not good* would be given additional weight (as a negative expression) but *not bad* would not.

3.1.5.2 Repetition Weighting

Any particular SO-valued word might appear a number of times within a text. Pang et al. (2002) noted that for a machine classifier, it was better to have a binary feature indicating the appearance of a particular unigram rather than a numeric feature that indicated the number of appearances. This does not seem to be true for word-counting models, nor necessarily should it be. Nevertheless, accounting for the psychological effects of a repeated word might be advantageous. In language, people often vary their vocabulary in order to avoid repetition, which can quickly lead to a kind of semantic bleaching.

(35) a. Overall, the film was excellent. The acting was excellent, and the plot was excellent.
b. Overall, the film was excellent. The acting was superb, and the plot was engrossing.

The repetition in (36a) has, I would argue, a weakening effect on overall strength of the impression that will be formed by the reader. Pragmatically, the repetition suggests that the writer lacks additional more substantive commentary, and is simply using a generic positive word. Another reason to tone down words that appear often in a text is that a word that appears regularly is more likely to have a neutral sense, for exactly that reason. This is particularly true of nouns. In one example from our corpus the words *death*, *turmoil*, and *war* each appear twice. A single use of any of these words might indicate a comment (e.g., *I was bored to death*), but repeated uses suggests a descriptive narrative. The current implementation of SO Calc allows for complete blocking of repetition, however the default opinion is to weight the nth appearance of a word so it has only 1/nth the SO value of the original; for example, the third *excellent* above would have an SO of 5 * 1/3 = 1.66. Repetitive weighting does not apply to words which have been intensified, with the rationale that the purpose of the intensifier is to draw special attention to them.

3.1.5.3 Position and XML weighting

It is clear that there are parts of a text that are more relevant to semantic analysis than others. Taboada and Grieve (2004) improved performance of an earlier version of the SO Calculator by weighting various parts of the text, putting the most weight on words at the two-thirds mark, and very little weight at the beginning. The current version has a simplified but user-configurable version of this weighting system; allowing any span of the text (with the end points represented by fractions of the entire text) to be given a certain weight.

An even more flexible and powerful system is provided by the XML weighting option. When this option is enabled, xml tag pairs in the text (e.g., <topic>, </topic>) will be used as a signal to the calculator that any words appearing between these xml tags should be multiplied by a certain given weight (with the weight specified in the configuration file; another option is to use the weight as a tag itself). This gives the calculator an interface to outside modules. For example, one program could preprocess the text and tag spans that are believed to be topic sentences, another program could provide discourse information such as rhetorical relations (Mann and Thompson 1988), and a third program could label the sentences that seem to be subjective. Armed with this information, the SO Calculator can disregard or de-emphasize parts of the text that are more relevant to sentiment analysis. In the next chapter, we will build a descriptive paragraph detector that makes use of this feature. At present, this interface only allows for weighting, but XML tags could be used to provide other kinds of information, for instance, the location of clause boundaries to improve the searching related to the linguistic features (negation, intensification, modality) discussed earlier in this chapter.

3.1.5.4 Multiple Cutoffs

Most work in sentiment analysis has been focused on binary positive/negative classification. Notable exceptions include Koppel and Schler (2005) and Pang and Lee (2005), who each adapted relevant SVM machine learning algorithms to sentiment classification with a three-class and four-class system, respectively. Since the SO Calculator outputs a numerical value that reflects both the polarity and strength of words appearing in the texts, it is fairly straightforward to extend the function of the SO Calculator to any level of granularity required; in particular, the SO Calculator grouping script takes a list of n SO cutoff values, and classifies texts into n+1 classes based on text SO values. The evaluative output gives information about exact matches and also near-misses (when a text is incorrectly classified into a neighboring class). This allows the SO Calculator to identify, for instance, the star rating that would be assigned to a consumer review.

3.2 A Quantificational Analysis of the SO Calculator

The focus of this Chapter so far has been showing how various features have been implemented in the SO Calculator and grounding those features in psychological/linguistic facts. However, the ultimate goal of the calculator is not to model the human faculty for interpreting affective language (which is undoubtedly much more complicated), but to provide a resource for the automatic analysis of text. As such, the accuracy of the program and the effect of its various features need to be evaluated.

For evaluation of the features, I use three corpora: the 400 texts of the Epinions corpora which were originally used as the basic for the dictionary, 1900 texts from the movie review Polarity Dataset (I have left out the 100 texts which were used to build our dictionary), the commonly-used collection of movie reviews discussed previously, and 3000 texts with camera, printer and stroller reviews, also from epinions.com. The latter two corpora (which we will call Movies and Cameras) were not used to build our dictionary or test the individual features of SO Calculator during development. For the majority of the results presented, we will use only 2400 of the Camera texts, since the other 600 are borderline 3-star reviews, which are typically avoided for the binary classification task. I include the Epinions corpus in part to show how similar the performance of "familiar" texts is to unknown texts. In machine learning, it is essential that training and testing be carried out on separate corpora, because a classifier will often learn to classify its training set too well, using features that are irrelevant to the actual task; for this reason, testing sets are usually set aside, or cross-validation used. However, this is not a concern for a semantic model, provided that the values are assigned to words based on their real-world prior polarity, and not the polarity of the text in which they appear (which is how SVM machine classifier learns its weights). Taking words from a corpus does (debatably) provide an advantage in terms of coverage, but provided the actual assignation of values is carried out independently, and the features justifiable in terms of general principle (and not just trying to maximize performance in a particular corpus), there is no danger of "overfitting" in the same way that machine classifiers are prone to. Instead, having a corpus where we have indentified most or all of the basic SO-valued words provides a useful testing ground for other, higher level features, which is anyway the primary concern of this work.

For the most part, our default configuration for the SO Calculator is as indicated in the previous section. Negative weighting and all phrase-level features are enabled. We begin with some basic statistics. Here, we use *intensity* to refer to the absolute SO value, i.e., the magnitude of the 1-dimensional vector.

	Corpus		
Feature	**Epinions**	**Movies**	**Cameras**
Total Number of Texts	400	1900	2400
Average Length, words per text	816	783	415
Average No. of SO-valued words per text	56.1	64.3	26.1
Percent positive/negative SO words	62/38	55/45	65/35
Percent noun/verb/adjective/adverb	18/14/59/8	26/25/51/8	19/14/57/9
Percent SO intensity of 1/2/3/4/5	39/30/19/7/5	36/32/22/7/4	41/32/15/6/4
Average SO word intensity	2.06	2.13	1.93
Average SO expression intensity	2.13	2.27	2.01
Average Text SO value	0.07	-0.23	0.31
Percent Correct Positive/Negative/All	86.0/74.5/80.3	63.4/89.4/76.4	90.5/69.8/80.2

Table 7: Basic SO Calc Statistics for the Three Corpora

To make sense of the numbers in Table 7, it is important to remember that the Epinions corpus has reviews of cultural products (e.g., movies and books) as well as consumer products (e.g., computers, phones), with substantially more of the latter. The differences between Movies and Cameras indicate the differences between the two sub-genres, with Epinions generally patterning after Cameras or falling somewhere in-between. When the length of text is controlled, Movies has more SO-valued words, more negative words, more intense words, more nouns and verbs, and poorer performance overall. We will explore this difference in more detail later (Chapter 4), for the moment we simply repeat the observation of Turney (2002) that movies (and other cultural products) differ from most consumer products in having a large amount of off-topic sentiment in the form of plot and character description.

Otherwise, though, the statistics in Table 7 are fairly uniform. Of note is support for *the Pollyanna Hypothesis* (Boucher and Osgood 1969), with positive words outnumbering negative words by a significant margin across all texts. Comparing different parts of speech, adjectives are the most relevant in terms of sentiment, significantly above the percentage that we would expect solely based on the numbers in the relevant dictionary (adverbs, on the other hand, are relatively rare). We also see that low SO words appear most often, with words of 2 SO being the most important when both frequency and intensity are taken into account; the average word intensity is also about 2. In general, the combined effect of features on individual word SO is to increase the intensity, but the overall effect is fairly minimal.

Table 8 gives information about the occurrence of various features.

	Corpus		
Feature	Epinions	Movies	Camera
Average Length, words per text	816	783	415
Average No. of Modifier Intensifications per text	7.64	6.85	4.14
Average SO Change per Intensification	0.87	0.99	0.73
Average No. of Negations per text	2.66	2.14	1.3
Average No. of Comparatives per text	1.25	0.85	0.65
Average No. of Superlatives per text	0.41	0.45	0.13
Average No. of Exclamations per text	1.76	0.81	1.09
Average No. of Capitalizations per text	0.28	0	0.23
Average No. of Highlights per text	2.83	3.07	1.21
Average No. of Verb/Modal Blocks per text	2.87	2.93	1.83
Average No. of Question Blocks per text	1.06	1.51	0.20
Average No. of Quote Blocks per text	0.77	1.22	0.17
Average No. of Imperative Blocks per text	0.31	0.27	0.21
Average No. of Modifier Blocks per text	0.28	0.69	0.10
Average No. of Negative Expressions per text	19.8	26.47	8.44
Average No. of Repeated SO words per text	11.9	9.88	5.31

Table 8: Average Feature Occurrence per Text

Again, there are some interesting contrasts among the various corpora; some features appear much more frequently in the Movies corpus, others in the Camera corpus. One feature that does not appear at all in the Movies corpus is capitalization, but this is simply because Polarity Dataset, as given, is entirely lower case. I speculate that some of the values for Movies corpus are indicative of informational-dense descriptions (for instance, more *but*, but fewer exclamations), but again, I leave a more detailed exploration to the next chapter. Overall, though, the occurrences of most of the features are fairly low, particularly when compared to the average number of SO-valued words per text from the previous table. Based on the numbers in Tables 7 and 8, we can postulate the average effect of each feature in a given text. For instance, in a 800 word text with an average word value of 2 SO, we would expect an average SO change of about 7 SO from intensification, 8-10 SO from negation (based on a 4 point shift), approximately 6 points each from irrealis blocking and highlighting, 20 SO from negative weighting, 18 SO from repetition blocking (though probably less, since low SO words are more likely to be repeated), and less than 5 SO for each of the various other features. With that in mind, consider Table 9:

	Corpus		
Feature	Epinions	Movies	Camera
Percentage of Texts within 2 SO of Cutoff	6.2	4.4	8.2
Percentage of Texts within 5 SO of Cutoff	16	10.2	19.9
Percentage of Texts within 10 SO of Cutoff	28.7	19.6	38.9
Percentage of Texts within 15 SO of Cutoff	38.7	30.2	55.2
Percentage of Texts within 20 SO of Cutoff	49	37.9	67.4

Table 9: Percentage of Texts within a Given Range of Cutoff Value (0)

The information in Table 9 shows that there are only a fairly small number of texts whose overall polarity can be affected by our various improvements. Even these numbers are far too optimistic, when three other facts are taken into account: 1) Of those texts whose SO values are near the cutoff, many of them are already being correctly identified; supposing we already have 80% accuracy, for instance, we would expect that only 20% of those borderline texts are actually incorrectly identified texts that are amenable to better SO calculation. 2) Longer texts are more likely to have more of the various valence shifting features, however changes to the SO of individual expressions have less far effect in this texts, generally, since Text SO represents an averaging of all SO values; in short, the texts that are likely to contain these features are less likely to be near the cutoff (this explains why the Camera corpus, which in general has shorter texts, has more texts near the cutoff). 3) Many texts have sentiment which is not being directed at the product under discussion, and thus correctly identifying low-level sentiment might not have a corresponding benefit at the level of the text, and sometimes the opposite will be true.

The upshot of all this is that text level performance is not necessarily the best way to evaluate whether these improvements are getting things right. As we will see in the rest of this section, for many of the improvements the numbers involved are quite small, below the level of statistical significance. However, this does not mean that we should disregard these kinds of details when doing sentiment analysis; on the contrary, as we increase our ability to detect what parts of the text should be taken into account, and which parts should be ignored, getting things correct "low-level" is likely to become more and more important.

We carry out evaluation using, as a baseline, the accuracy of SO Calc with all features enabled and at default settings, and then disabling features or modifying settings to see the effect on accuracy. Each table will contain the baseline accuracy, for ease of comparison. Accuracy is calculated as a percentage: the number of texts correctly identified as positive or negative divided by the total number of texts. Results which are statistically significant ($p<0.05$, using a chi-squared test) as compared to the baseline (taken as the expected value) will be marked with an asterisk (*). In addition to the results for each individual corpus, I also provide a combined result, treating all 4700 texts as a single corpus. It should be noted, however, that this number is biased towards the Camera corpus, since it contains more texts.

The first set of results in Table 10 is focused on testing the various dictionary combinations. Google refers to the adjective-only dictionary build using the SO-PMI method, as described in

Section 1. The Subjectivity dictionary is based on the multi-POS list of subjectivity cues of Wilson et al. (2005) derived from both manual and automated sources; weak positive/negative cues were given SO values of +2/-2, and strong positive/negative cues values of +4/-4. The simple dictionary refers to the main SO Calc dictionary, but with all SO values simplified to +2/-2, with +1/-1 intensification. Otherwise, nouns, verb, adjectives, and adverbs refer to the corresponding main SO dictionary—double weight adjectives refers to putting twice the normal weight on adjective-based expressions.

	Accuracy by Corpus			
Dictionary	**Epinions**	**Movies**	**Cameras**	**Combined**
Full	80.25	76.37	80.16	78.64
Google	62.00*	66.31*	61.25*	63.36*
Subjectivity	67.75*	62.89*	70.79*	67.34*
Simple	76.75	69.79*	78.71	74.93*
Full w/o Nouns	78.5	77.05	76.96*	77.12*
Full w/o Verbs	78.75	75.84	77.75*	77.06*
Full w/o Adjectives	72.25*	64.31*	72.58*	69.21*
Full w/o Adverbs	78.75	73.95*	79.16	77.02*
Only Adjectives	72.25*	76.63	71.98*	73.92*
Full, Double Weight Adjectives	78.75	76.89	77.54	77.38*
Full w/o Multi-word	80.75	75.68	79.54	78.08

Table 10: Accuracy with Various Dictionary Options

Overall, our full dictionary has the best performance, significantly better than almost all other options when the corpora are considered together. Dictionaries created automatically, either full or in part, do poorly compared to manually created dictionaries. One interesting result is how verbs play a marginal role and nouns a slightly negative role in the calculation of the Movies dictionaries, probably because they are overused in the calculation (see table 7); again, this can be attributed to the presence of unreliable plot description. In the Camera corpus, however, nouns and verbs both significantly contribute to performance, indicating that they should not be ignored. Although adjectives are clearly the most important part of speech, increasing the weight on them does not improve overall performance. In the movies corpus, adverbs play a surprisingly important role, despite their underrepresentation. The effect of multiword expressions, on the other hand, is quite modest; the current dictionaries only have a small, preliminary collection of them, however, so their minimal influence is not altogether surprising.

Next, we look at intensification.

	Accuracy by Corpus			
Options	**Epinions**	**Movies**	**Cameras**	**Combined**
Full	80.25	76.37	80.16	78.64
No (Modifier) Intensification	78.75	75.44	78.33	77.20*
No Comparatives/Superlatives	78.25	75.52	80.00	78.04
No *but* Highlighting	80.25	74.68	79.37	77.55
No Capitalization Intensification	80.25	76.37	80.16	78.64
No Exclamation Intensification	80.00	76.42	79.89	78.50
1.5 Weight on Intensified Words	78.75	76.00	80.05	78.30
10 Weight on Intensified Words	74.50*	71.31*	73.75*	73.83*
1.5 Weight on Capital/Exclam[6]	80.50	76.31	80.04	78.57
2.5 Weight on Capital/Exclam	80.50	76.47	79.87	78.55

Table 11: Accuracy with Various Intensifier Options

Table 11 indicates that basic modifier intensification provides a significant boost when considered across all corpora. The effect of the other features, however, is not so clear cut. Capitalization intensification, for instance, does not affect performance at all, not altogether a surprise giving its extreme rarity in relevant texts (see Table 7); exclamation intensification helps, but minimally. The effect of *but* highlighting (which falls just short of statistical significance) suggests that using discourse markers is a good way to find more relevant information. Despite their inherent subjectivity, comparatives and superlatives are somewhat helpful.

I also tested two parameters related to intensification: the weight applied to intensified expressions, and the SO modifier for words where capitalization/exclamation intensification applied (the default is 2). Increasing the value of intensified expressions to a rather ridiculous degree had improved performance in an earlier version of the SO Calc, however the effect has disappeared (perhaps as the result of better negative weighting). Increasing the value of the modifier on Capitalization/Exclamation improved performance slightly in the Movies corpus, but the overall effect was negative.

Table 12 presents results relevant to negation. Recall that there are three methods for a negation: shift (3->-1), flip(3->-3), and shift limited by flip (3-> -1, 1-> -1), and the backwards search for negators can be restricted to a small set of words/tags, or only blocked by a boundary. For shifts, it is possible to set a different shift value for each POS, by default adjectives/adverbs is 4, and nouns/verb is 3 (because the SO value of the latter tends to be lower). Here, we tested several other possible values.

[6] i.e., capitalization and exclamation intensification

	Accuracy by Corpus			
Options	**Epinions**	**Movies**	**Cameras**	**Combined**
Full (shift negation, all restricted, adj/adv[7] shift 4, noun/verb shift 3)	80.25	76.37	80.16	78.64
No negation	75.75*	74.31*	76.12*	75.36*
Flip negation	80.00	75.57	80.04	78.23
Limited shift negation	79.50	76.05	79.41	78.06
Unrestricted negator search (all)	78.50	74.42	80.04	77.60
Unrestricted negator search (verbs)	79.75	76.21	80.46	78.68
Shift value 4 for noun/verb	80.25	76.05	80.50	78.68
Shift value 3 for adj/adv	79.75	76.31	79.42	78.19
Shift value 5 for adj/adv	79.75	75.58	80.66	78.68

Table 12: Accuracy with Various Negation Options

Negation does provide a significant performance increase, more so than intensification. Full shift negation does seem to be preferable to the other options; though the results are not significant (nor should they be, with 1.3-2.7 negations per text and an average difference of less than 2 SO between methods), they are, however, consistent across the three corpora. The testing of restricted versus unrestricted negation found that restricted negation was a somewhat better option, though for verbs, unrestricted negation is slightly better; this could easily be an anomaly, or it might reflect the fact that unrestricted negation captures negation-raising effects. With regards to shift values, it is also difficult to come to any firm conclusions because the behavior of the shift is clearly tied to the positive and negative biases of the texts; positive biased texts like Camera do better when negative shifting is increased (which, on average, will lower text SO, since positive words, being more common in general, are more likely to be negated), whereas the opposite is true in Movies (which is negatively biased after the 1.5 negative weighting). The slight preference for a shift verb/noun 4 can, in this case, be attributed to the greater number of texts in the camera corpus. There is thus no strong evidence for changing the default shift value, though we might get some improvement in individual domains by tinkering with the values.

We turn now to the performance of the various irrealis blocking features.

[7] i.e., adjective/adverb

	Accuracy by Corpus			
Options	Epinions	Movies	Cameras	Combined
Full (Modifier Block at Intensity 3)	80.25	76.37	80.16	78.64
No Modal/Verb/NPI[8] Blocking	79.75	75.95	79.25	77.95
No Question Blocking	79.25	75.84	80.37	78.44
No Quote Blocking	79.25	75.73	80.08	78.29
No Imperative Blocking	80.25	76.37	80.32	78.71
No Definite Unblocking	80.50	76.37	79.79	78.46
No Modifier Blocking	80.00	75.87	80.12	78.39
Modifier Block Intensity = 2	80.25	76.10	80.21	78.53
Modifier Block Intensity = 4	80.00	75.89	80.12	78.40
Disable All Non-Significant Features	77.75	73.53*	78.25*	76.29*

Table 13: Accuracy with Various Irrealis Options

Most of the features here appear barely once per text (less in the Camera corpus). The main irrealis feature, modal/verb/NPI blocking, is more common, and this is reflected in consistent (though not significant) performance improvement among the corpora. Imperative blocking actually degrades performance, which might reflect the unreliability of imperative form in informal texts. Quote and modifier blocking have a slightly positive influence on performance in all corpora; the results also indicate that 3 (the SO value of *too*) is the right intensity cutoff for modifier blocking, having either more (2) or less (4) modifier blocking both lower performance. The final result reported in Table 13 is the result of disabling all features with statistically insignificant effects from Tables 11and 13. The combined effects of these features *is* significant, with a contribution to the overall performance that is greater than (basic) intensification, but not as large as negation. Although there are no "silver bullets" here, there is a cumulative benefit to these minor improvements.

Table 14 reports the results of testing text level weights and cutoffs (i.e., the value which serves as the boundary between positive and negative text classification, by default it is 0). There are two see-saw battles here: the first is the information value of some repeated words versus the semantic noise associated with others. As it turns out, it is better to throw out all repeated words than it is to use them with full SO, but the better (perhaps not yet the best) solution is to balance the two, using only part of the information, as we have here with 1/N. Interestingly, in the high-coverage Epinions corpus, repetition weighting is not helpful. Otherwise, the two larger test corpora show almost identical performance with respect to this feature, which suggests that it is fairly robust.

[8] i.e., negative polarity item (e.g., *any*)

	Accuracy by Corpus			
Options	Epinions	Movies	Cameras	Combined
Full (Negative Weight = 1.5, 1/N Repetition Weighting)	80.25	76.37	80.16	78.64
No Repetition Weighting	81.5	75.08	79.15	77.70
Full Repetition Blocking	78.75	75.79	80.08	78.24
No Negative Weighting (=1)	71.25*	75.63	71.71*	73.25*
Negative Weight = 1.25	78.00	76.99	76.75*	76.95*
Negative Weight = 1.4	81.25	77.1	79.04	78.45
Negative Weight = 1.6	78.75	73.68*	81.04	77.87
Negative Weight = 1.75	78.25	71.99*	80.83	77.03*
No Negative Weight, Cutoff .3	81.25	76.94	79.20	78.47
No Negative Weight, Cutoff .35	80.75	76.73	80.21	78.85
No Negative Weight, Cutoff .4	79.75	75.52	79.91	78.13

Table 14: Accuracy with Various Text Level Options

Negative weighting, on the other hand, results in conflict between maximizing performance in one large corpus rather than the other. Both corpora benefit from negative weighting, but, as reflected in the original differences in positive/negative word ratio, the two corpora benefit to different degrees (the Movies corpus less, the Camera corpus more). Either weighting or the changing of cutoff value can be used to find the optimal balance for one, or the other, or both together. As it happens, a cutoff of .35 provides slightly better performance than a negative weight of 1.5, but in fact it is possible to fiddle with either feature (or both together) to get slightly better overall performance. There is little value to this, however, since we would simply be maximizing performance for this particular combination of texts. It would be more useful to derive values that generally work well for a given (sub-)genre; we could improve performance by identifying the genre of the text, and then using the best settings for that genre, an option we will explore in the next chapter; here, a negative weight of 1.4 improves performance for the movies reviews, and a weight of 1.6 works wells in camera reviews.

The final part of this chapter will look at the multi-class case. For this, we will make use of the full 3000 text Camera corpus, which includes 600 texts which were assigned a star rating of 3 (the exact middle of the scale). Because the website allows users to give provide a star rating and a yes/no recommendation, we were able to maintain an equal balance between yes/no reviews (300 of each). However, our performance identifying the recommendation of these reviews is awful: 56.7% (barely above chance), which suggests that they do not fit well into the positive/negative two-class schema we have been assuming thus far.

In order to do multi-class classification, we need a new method for determining cutoff value (0 will obviously not work). For simplicity, I simply select cutoff values that divide the space into regions with approximately equal numbers of texts. Since we know that in this case there are in fact equal numbers of each rating, this is probably the optimal method. In cases where the number of texts of each rating aren't likely to be equal (i.e., the real world), the best approach would probably be to derive a distribution using random sampling, and select cutoff values based on that.

We will test the SO Calculator on 3 tasks: the 3-class task, the 4-class task, and the 5-class task. We would predict that these tasks are likely to get progressively more difficult, as the random baseline gets progressively lower (33%, 25%, 20%). For the 3-class task, we use 1-star (negative), 3-star(ambivalent), and 5-star (positive) reviews, for the 4-class we use 1-star (strongly negative), 2-star (weakly negative), 4-star(weakly positive), and 5-star reviews (strongly positive), and for the 5-class task we use all 3000 reviews. The confusion matrix for the 3-class task is given in Table 15.

	Classified as			**Accuracy**	
Class	**Negative**	**Ambivalent**	**Positive**	**Exact**	**Close**
Negative	419	142	39	69.8	93.5
Ambivalent	155	291	154	48.5	100.0
Positive	26	167	407	67.8	95.6
All	600	600	600	62.1	96.4

Table 15: Confusion Matrix and Accuracy for the 3-Class Task

Recall that "close" refers includes both the correct class as well as neighboring classes; the high close value indicates that there are very few catastrophic misclassifications (where a positive text is labeled as a negative text), a promising result. Though only about 50% of the ambivalent texts are correctly labeled, the rest are split equally between positive and negative, exactly what we would expect considering their original makeup. Overall, performance is quite acceptable compared to a 33% random baseline; there is little doubt that the SO Calculator is able to distinguish between these 3 classes to a significant extent. Next, we look at the results from the 4-class task:

	Classified as				**Accuracy**	
Class	**S-Neg**	**W-Neg**	**W-Pos**	**S-Pos**	**Exact**	**Close**
Strongly Negative	346	177	53	24	57.7	87.2
Weakly Negative	208	242	104	46	40.3	92.3
Weakly Positive	28	109	239	224	39.8	95.3
Strongly Positive	19	73	205	303	50.5	84.7
All	601	601	601	597	47.1	89.9

Table 16: Confusion Matrix and Accuracy for the 4-Class Task

Accuracy has dropped, but the overall pattern remains the same: the vast majority of texts are classified to the correct class or an immediately neighboring class. We can see the effects of the missing ambivalent class here; far more W-Pos/Neg texts are classified as S-Pos/Neg rather than W-Neg/Pos, indicating that the two positive (negative) classes are closer in semantic space. Very few strong texts are classified as strong texts of the opposite polarity, indicating their distance. Finally, we look at the full 5-class case:

	Classified as					**Accuracy**	
Class	**S-Neg**	**W-Neg**	**Ambiv.**	**W-Pos**	**S-Pos**	**Exact**	**Close**
Strongly Negative	314	169	67	31	19	52.3	80.5
Weakly Negative	173	202	131	52	42	33.7	84.3
Ambivalent	80	142	184	122	72	30.6	74.6
Weakly Positive	20	61	143	181	195	30.2	86.5
Strongly Positive	13	26	116	173	272	45.3	74.2
All	600	600	641	559	600	38.4	80.0

Table 17: Confusion Matrix and Accuracy for the 5-Class Task

There was a clustering of values near the Ambivalence/W-Pos boundary, leading to a slightly unbalanced distribution in the 5-class case; this, however, does not affect the overall outlook, which is again fairly good. Across the three tasks, accuracy drops, predictably, but seems to be fairly steady at a little less than twice the expected (random) performance, which is more than enough for extremely high statistical significance (p <0.0001). As we increase the number of classes, the value of our Close index stays fairly high, indicating that we can be fairly confident that the classification given by SO Calc is not seriously in error, particularly at the edges of the spectrum.

Abstracting away from the particular cutoff values we have chosen, Figure 2 shows the range, mean, and best-fix line for the SO values of the texts, divided according to their original class.

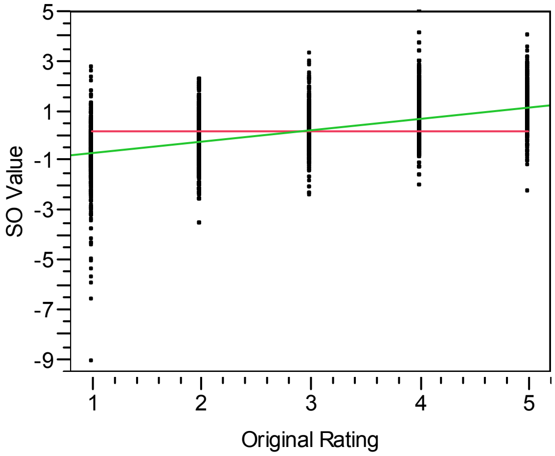

Figure 2: SO Value versus Original rating, 5-Star Classification

The differences in the means for all the various ratings are significant (p<0.0001), and the R-Squared value for the best of fix line is .35, suggesting that 35% of the variation in SO value can be explained by the rating; there is clearly a lot of other variation going on as well. Interestingly, trying to get a best-of-fit with higher degree polynomials does not change the picture much, indicating that relationship between SO value and original rating is, in fact, linear. Positive and negative are not entirely symmetrical, however: at the negative end of the spectrum, it is possible to be 100% confident that a review is strongly negative(SO < -4) , however, this is not true at the positive end, where a 4-star review actually has the highest rating. In general, the variation in SO value for strongly negative reviews is higher than any other category.

Pang and Lee (2004) include a small study to test whether humans could distinguish between reviews of different star ratings. Not surprisingly, when the difference was high (2-3 stars) humans have near perfect performance, but even humans have difficulty deciding if two texts have exactly the same star rating (an average performance of 55%, though the scale they used involved half-stars). Thus we have reason to be optimistic about multi-class classification with the SO Calculator; though having an SO value is not conclusive, it would provide a most likely choice and probabilities for other possible options.

Having discussed the theory behind and implementation of various features of the SO Calculator, including an evaluation of relevant features, in the next chapter we will look an external module that further boosts performance.

Chapter 4: Genre Classification

As described in the previous chapter, the SO Calculator includes a simple yet highly flexible mechanism for interacting with external modules, namely weighting of the SO words appearing in XML-enclosed text spans. In this chapter, I will exploit this feature, building machine learning classifiers to automatically detect paragraphs within movie reviews which are more or less likely to contain relevant sentiment. For paragraph-level genre classification, I will examine two approaches: one which uses the SO Calculator itself to get information that a classifier can use to train, and one which depends on annotated data. Finally, I will integrate a layer of text-level sub-genre classification that allows us to optimize the use of negative weights.

4.1 Background

In the literature, genre has primarily been defined in terms of a communicative purpose and the stages which are necessary to achieve that purpose (Eggins, 2004, Martin, 1984). Reviews, where the overall goal is to provide opinionated information about a product in the public domain, are clearly an example of a genre. Like all genres, reviews are made up of a series of stages; importantly, reviews in different domains call for different stages (e.g., movie reviews generally involve plot and character description, for instance, whereas appliance reviews do not), and this fact allows us to further divide the review genre into various sub-genres. Crucially, if we also view a review as potentially an example of a macrogenre (Martin, 1992), a genre which contains instances of other genres, then the difference between stages contained in these reviews can also justifiably be referred to as a distinction of genre; after all, the goal of providing descriptive information and the goal of communicating opinion are quite distinct, and, as we will see, are often (though certainly not always) addressed independently in a review. In what follows, we will be interested in genre on both levels, since both sorts of information will allow us to improve the performance of the SO calculator, either by using genre-specific configuration with SO Calc or by disregarding (or discounting) SO-valued words in paragraphs whose overall purpose is not, apparently, to communicate sentiment.

The approach taken here is directly influenced by Finn and Kushmerick (2003), who approached the detection of subjectivity and polarity in text as a type of genre classification (though the latter, at least, would probably not meet a more formal definition of genre). In building decision tree classifiers to detect the polarity of various types of product reviews, they tested three main sets of features, namely unigram bag-of-words (BOW) features, part of speech features, and text statistics, which includes features related to word, sentence, and text length, as well as function word and punctuation frequency. For subjectivity detection, a mixed model worked the best, but the BOW performance dropped sharply across domains. Polarity was more difficult to classify: BOW worked fairly well, but only within a single domain, with the decision tree splitting early on features like the name of actors appearing in bad movies.

Pang and Lee (2004) was aimed at improving the performance of an SVM sentiment classifier by identifying and removing objective sentences from the texts, training a unigram subjective/objective sentence classifier on review/plot snippets. They applied graph theory to minimize the number of subjective/objective switches throughout the texts, removing those sentences which were found to be objective (descriptive). Results were somewhat mixed: the improvement after objective sentences were removed was minimal for the SVM classifier (the performance of a naïve Bayes classifier, though, was significantly boosted), however testing with *only* parts of the text classified as objective showed that the parts that had been eliminated were indeed mostly irrelevant. They reported a drop in performance when paragraphs were taken as the only possible boundary between subjective and objective spans. Sentence and phrase-level subjectivity detection has received a great deal of attention, though primarily in the domain of newspaper articles (Wiebe et al., 2003, Wilson et al., 2005, Yu and Hatzivassiloglou, 2003).

Other research that has dealt with identifying more or less relevant parts of the text for the purposes of sentiment analysis include Taboada and Grieve (2004), who improved the performance of a word-counting model by weighing words towards the end of the text, and Voll and Taboada (2007), who used a topic classifier and discourse parser to eliminate potentially off-topic or less important sentences.

Bieler et al. (2007) are concerned with identifying formal and functional zones within the genre of the movie review, which we will focus on in the next two sections. Formal zones are parts of the text which are entirely characteristic of the genre, for movie reviews this includes basic information about the movie (e.g., the title, cast, etc.) and the review (e.g., the author, date of publication, etc.); functional zones, on the other hand, serve as the main content of the review, and can be roughly divided into two types, describe and comment. Bieler et al. showed that the functional zones could be identified fairly successfully using 5-gram SVM classifiers built from an annotated German corpus. I will use their describe/comment terminology for the rest of the discussion, with the idea that describe/comment genre distinction essentially captures the objectivity/subjectivity distinction in the domain of movie reviews. Subjectivity is a fairly broad and nebulous concept, note for instance that the plot description below would probably not be considered objective in another context.

(37) The movie opens up with Nathan Algren (Tom Cruise) sabotaging his own employment opportunities by funneling his anger and regret into a bottle of alcohol and an attitude of rebellion. What's he so upset about? As a captain, he spent time fighting those savages and along the way, he massacred innocent women and children. That's enough to make any man with a conscious, drink.

Words like *sabotaging*, *massacred*, *regret*, *innocent*, and *savages* seem inherently subjective, however in this case they are used to describe the actions of a fictional character, and as such reflect the world as it is presented in the film, and not the opinion of the reviewer.

In the preceding chapter, we noted some major statistical differences between movie reviews and other product reviews; before moving on to the building of classifiers, we look at some of the numbers in more detail. Table 18 shows the percent accuracy of the SO calculator using different negative weights (extra weight applied to all negative SO expressions in the text) for the two large, homogeneous corpora, the Polarity Dataset (2000 movie reviews) and the 2400 text Camera corpus. Pos, Neg and All indicate performance on positive, negative, and all texts, respectively.[9]

Neg Weight	Movies			Camera		
	Pos	Neg	All	Pos	Neg	All
1.2	77.4	76.4	77.1	93.7	58.4	76.0
1.3	73.2	82.4	77.8	91.7	62.8	77.9
1.4	68.7	86.2	77.5	91.7	66.2	78.9
1.5	62.8	89.7	76.0	90.4	70.2	80.3
1.6	56.8	91.2	74.0	89.4	72.5	80.9
1.7	53	92.4	72.7	87.4	74.7	81.1
1.8	49.1	93.7	71.4	85.0	77.0	81.0

Table 18: Accuracy by Corpus and Negative Weight

Negative weights higher than 1.8 or lower than 1.2 lead to progressively worse performance on both corpora. Two facts stand out: First, movie reviews are indeed more difficult that other product reviews, with a 3.3% difference in maximum accuracy even when the optimal negative weight for each corpus is used. Second, although both domains have positive bias, movie reviews have a much lower tolerance for negative weighting, and our optimal weight of 1.5 is actually a compromise between the ideal weights for movies (1.3) and cameras (1.7). This suggests that we can improve performance using genre detection at the level of text. First, however, I will show how the performance gap between the two types of texts can be narrowed using an external module.

4.2 Semi-Supervised Genre Detection

4.2.1 Preliminary Investigation

As we have seen, the SO calculator can identify the polarity of entire texts with reasonable accuracy, and it follows that it could also be used to classify smaller units within a text. For our existing corpora, we already know the correct polarity of each text, as it was extracted automatically from the original html. These two facts, taken together, allow for a novel approach: I can automatically identify paragraphs that apparently conflict with the overall

[9] The discrepancies in the movie review numbers as compared to those presented the previous chapter are reflections of two facts: for this task, I have added back the 100 reviews from the Polarity Dataset that were used in the creation of the SO Calc dictionary. Also, I use a version of the Polarity Dataset which was extracted directly from the original html, preserving paragraph breaks (which are not in the standard corpus) and capitalization. Note that these changes have actually resulted in a slight (.3%) drop in performance.

sentiment of the text, and these inconsistent paragraphs can then be used to train a classifier to detect paragraphs that should be discounted by the SO calculator (using XML weighting). With this in mind, I extracted paragraphs from the original html source code for the Polarity Dataset (the official version did not conserve paragraphs breaks), ran each paragraph through the SO calculator, and classified them by the polarity of the text within which they appeared as well as whether they were consistent or inconsistent with respect to the polarity of the text, or just neutral (no SO value). For this purpose, I chose a negative weight (1.22) for the SO calculator that resulted in practically equal positive and negative accuracy, so that the SO Calculator positive/negative text counts (and thus presumably the positive/negative paragraph counts) were somewhat balanced. Altogether, there were 14,786 paragraphs, including 4,489 positive consistent, 2,667 positive inconsistent, 3,987 negative consistent, 2,272 negative inconsistent, 732 positive neutral, and 639 negative neutral.

The initial hypothesis was that I could train a binary classifier on consistent and inconsistent texts, however this quickly proved to be unworkable: When I first examined the features (discussed in detail later), there were far too many differences between inconsistent paragraphs in positive and negative texts. To investigate this, I carried out a small corpus study, randomly extracting 100 samples for each type of paragraph, and categorizing using the basic scheme proposed by Bieler et al. (2007). The describe tag was reserved for paragraphs involving only plot, character, or general description of the movie content, whereas paragraphs that clearly indicated an attitude were tagged as comment (even if they also contained description). Each comment was also checked for whether the SO Calculator had accurately determined its polarity. The results are given in Table 19.

Type	Formal	Describe	Comment	Accurate
C-Pos	2	27	70	68
C-Neg	2	21	77	75
I-Pos	5	54	41	14
I-Neg	3	21	75	14
N-Pos	76	17	7	0
N-Neg	72	6	22	0

Table 19: Tags of Sampled Paragraphs[10]

Accurate refers to the number (not percentage) of comment paragraphs where the polarity of the paragraph as derived by the SO Calculator was the same as the polarity judged by the human annotator (myself). The fact that accuracy was fairly low for both inconsistent paragraphs means that the majority of inconsistent paragraphs weren't inconsistent because they were negative comments included in positive reviews (or vice versa), but rather they were inconsistent because the SO calculator had failed to identify the polarity correctly. Consistent

[10] C = consistent (paragraph SO as determined by SO Calc has same polarity as the text), I= inconsistent (paragraph SO is opposite of text SO), N = Neutral (paragraph SO is 0). Pos and Neg refer to the known polarity of the text in which the paragraph appears.

paragraphs, whether positive or negative, were fairly uniform, being mostly accurate comment with a small percentage of description and a very small amount of formal or inaccurate comment. For inconsistent, texts, however, there is a large discrepancy between the type counts: inconsistent positive texts seem to be mostly description (54%), whereas the majority of inconsistent negative texts sampled (61%) were comment that had been incorrectly handled by the SO calculator. The latter often involves mixed comment and description, words not the dictionary, complex linguistic structure, missed discourse cues, or off-topic asides. In general, positive texts had far more description than negative texts, as evidenced even in the neutral samples (which were otherwise mostly formal zones). This perhaps explains why positive text performance drops so quickly for movie reviews: Most positive texts will contain description, which inevitability involves antagonists, dilemmas, and conflict, and when the weight on those elements in increased, the positive texts are misclassified. Positive reviews in other domains, however, might have few if any negative words to begin with.

Noting the significant differences between the various paragraph types, I proceeded with the preliminary training of a machine classifier. My first attempt involved a 4-way classification between formal (represented in the dataset by neutral reviews) comment (represented by consistent reviews), describe (represented by inconsistent positive reviews), and “inaccurate” (represented by inconsistent negative reviews). My hope was that I could identify anything that was either description, a formal zone, or a comment that would be difficult for the SO Calculator to classify properly, and discount it appropriately. To build the classifiers, I used the WEKA data mining suite (Witten and Frank, 2005), initially testing with C4.5 decision tree algorithm, a naïve Bayes classifier, and a SVM classifier; it became clear, however, that the SVM classifier was far outperforming the other two (by a margin of about 5% as tested on our manually-tagged data), and, given SVM classifiers proven aptitude in sentiment-related classification (Pang et al., 2002), I did not pursue the others further. A major drawback to using an SVM, however, was its tendency to favor the most common class; when I used the full dataset (which was over 50% consistent paragraphs), the SVM would only classify paragraphs as comment or formal, ignoring the noisy describe and mixed classes altogether. This makes sense in the context of Table 2, since when there is twice as many examples of consistent paragraphs (positive or negative) as inconsistent positive paragraphs, the two classes actually contain approximately equal instances of description, a circumstance that would probably confound the classifier. In order to get more interesting results, I just equalized the counts of the three larger classes, throwing out a lot of comment in the process; formal zones were apparently distinctive enough so that their lower count could be maintained. I used a small set of 24 promising features, many of the same features that were used in our final model (see Section 5), and trained on about 5000 instances. The results of our best classification using this rubric are given in the confusion matrix below (Table 20). The numbers are instances from our manually-tagged test set from Table 19.

Type	Classified as			
	Comment	Describe	Formal	Inaccurate
Comment	216	32	3	0
Describe	46	82	5	0
Formal	6	4	140	0
Inaccurate	85	24	11	0

Table 20: Classification of Test Set with Inaccurate

The classifier simply refused to label anything as “inaccurate,” suggesting that there is no easy way to distinguish comment that the SO Calc can properly classify from that which it cannot. At the other end of the spectrum, the formal zones are easily classified based on a small set of features, since they tend to lack verbs, and are predominantly proper nouns and punctuation; we will not discuss their classification further. Most interesting, of course, is the classification of comment and describe; both precision and recall for comment and describe are well above 50%, indicating that they can be distinguished to some extent. In the next section, we look closely at features that distinguish comment from describe and tie them to an existing framework, the genre analysis of Biber (1988).

4.2.2 Feature Selection

One drawback of using the SO Calculator to group paragraphs is that caution must be exercised with respect to the features used for automatic classification; relying on a pure bag of words approach would almost certainly result in a machine classifier that simply makes use of the same words as the SO Calculator originally did. I have opted instead to focus on a small set of features whose reasonableness for the task at hand I can analyze; in the next section I will evaluate N-gram classifiers built on annotated texts.

Biber (1988) used a compact but wide-ranging set of features to characterize the tendencies of various written and spoken text genres. Since the distinction between description and comment could indeed be viewed as one of genre (as discussed in Section 1), this seemed to be a good place to begin. Instead of looking at individual words, Biber grouped words into categories based on their discourse function, e.g., first, second, and third person pronouns, demonstrative pronouns, place and time adverbials, amplifiers and downtowners, hedges, possibility, necessity, and predictive modals, etc., mostly derived from relevant word lists in Quirk et al. (1985a).To capture certain categories it was only necessary to use part of speech information provided by a tagger (Brill, 1992), but I also created a number of word lists, including multi-word expressions. For amplifiers, downplayers, and quantifiers, I used words from the SO Calculator intensifier dictionary that were not inherently positive or negative.

With respect to discourse markers, Biber’s classification scheme was not detailed enough for my purposes, so I used cues from Knott (1996) to add several additional categories, including closed-class words and expressions that indicate contrast, comparison, causation, evidence, condition, introduction, conclusion, alternatives, topic continuation, and topic change. Examples under the comparison heading, for instance, include *more*, *in comparison*, *equally*, and several other expressions and tags. Though this rubric includes a wide variety of discourse categories, some of the categories from Rhetorical Structure Theory (Mann and Thompson, 1988) have been collapsed to avoid ambiguity; for instance, contrast, antithesis, and concession have been collapsed in contrast, so the discourse marker *but* is only under one category rather than three.

I also included three sets of about 500 adjectives, categorized not by polarity but by appraisal group, i.e., into Appreciation, Judgment, and Affect (Martin and White, 2005), as discussed in Section 2 of Chapter 1. This information has been shown to be useful in improving the performance of polarity classifiers (Whitelaw et al., 2005), and also seems very relevant to the distinction being drawn here; it is easily to see how Judgment, for instance, might be used more in character description. Note that the word lists are fairly evenly balanced between positive and negative words, so there is no danger of these features reflecting positive or negative polarity.

In addition, I added some basic text statistics, including the length of words and sentences, and three features (two binary first/last features, and one numerical 0-1 range) indicating the position of paragraph in its original text. Two domain specific features were used that captured the collocation of parentheses and proper nouns, which are very common in descriptive passages where the name of an actor is mentioned in parentheses.

Finally, I also looked at a neutral subset of words used to identify subjectivity at the sentence level in newspaper articles (Wilson et al., 2005), however I quickly decided that the words not already included in elsewhere in our analysis were, for the most part, unlikely to appear or somewhat newspaper-genre specific. A few examples, pulled from the A's, are sufficient to demonstrate my point: *activist, adolescents, alert, alliance,* and *appearance* are, in the domain of movie reviews, just as likely or perhaps even more likely to be used in description rather than comment, and including essentially random words like these would be little different than an n-gram approach (see the next section).

All together, I began with 74 main features covering 1000+ words and expressions; though I extracted information for each word/expression, the classifier ultimately trained on the frequency of *classes* of words, not the individual words, i.e., the texts did not have features for *excellent* and *terrible*, only a feature for *Appreciation* which was calculated based on the appearance of any of the words in the Appreciation list (including both *excellent* and *terrible*). As such, there was no way that the machine classifier could simply mimic the SO Calculator, which works entirely based on the SO value of individual words.

Except for certain textual features already noted, I followed Biber in normalizing the frequency of our features to appearances per 1000 words. I briefly tested binary features, but they seem to result in less significant differences among the various classes. To carry out feature selection, features were extracted from each of the 14,786 paragraphs, the mean and standard deviation calculated for each type, and then t-testing conducted to determine those features which showed significant differences across paragraph class. As suggested by the results in Table 20, there were few differences (mostly trivial) between negative consistent and negative inconsistent paragraphs, and an excess of differences between neutral and other types. Below, I have focused on the differences between positive consistent (mostly positive comment), negative consistent (mostly negative comment), and positive inconsistent (mostly description) paragraphs, as there were a number of interesting variations observed.

Biber (1988) included a factor analysis using relevant features to determine certain textual "dimensions." Three of his dimensions are directly visible in the data. The first, and most significant, is a dimension which marks "high informational density and exact informational content versus affective, interactional density" (107). The difference and p-value for the significant differences of some relevant features are given in Table 21.

Feature	Difference	P-value
Exclamations	C-Neg > I-Pos	8.0 *10^{-5}
	C-Neg > C-Pos	7.0 *10^{-7}
1st person pronouns	C-Pos > I-Pos	.003
	C-Neg > C-Pos	.0006
	C-Neg > I-Pos	2.8 *10^{-9}
Contractions	C-Pos > I-Pos	.0006
	C-Neg > C-Pos	7.1 *10-^{14}
	C-Neg > I-Pos	1.1 *10^{-19}
Intensifiers	C-Pos > I-Pos	6.3 *10^{-13}
	C-Pos > C-Neg	2.1 *10^{-5}
	C-Neg > I-Pos	.0005
Word Length	C-Pos > C-Neg	1.9 *10^{-11}
	I-Pos > C-Neg	2.3 *10^{-12}
Place Adverbials	I-Pos > C-Pos	1.7 *10^{-6}
	C-Neg > C-Pos	2.0*10^{-5}

Table 21: Feature P-values for Genre Dimension 1

Other relevant features that showed significant differences include 2nd person pronouns, demonstrative pronouns, pronoun it, hedges, WH-clauses, and adverbs. It is fairly clear that, in general, descriptive paragraphs (I-Pos) are at one end of the spectrum (the informational end) while negative comments (C-Neg) are at the other (the affective end); positive comment (C-Pos) tends to fall somewhere in-between (but not always, see, for instance, place adverbials).

The second dimension distinguishes narrative discourse from other types of discourse. Key features and p-values are given in Table 22.

Feature	Difference	P-value
3rd Person Pronouns	I-Pos > C-Pos	$2.3 * 10^{-16}$
	I-Pos > C-Neg	$9.4 * 10^{-16}$
AUX + VBN[11]	I-Pos > C-Pos	$2.4 * 10^{-9}$
	I-Pos > C-Neg	$4.6 * 10^{-5}$
Time Adverbials	I-Pos> C-Pos	$1.3 * 10^{-8}$
	I-Pos>C-Pos	$7.8 * 10^{-5}$

Table 22: Feature P-values for Genre Dimension 2

Biber notes that narratives are usually characterized by a predominance of past tense verbs, however plot description in movie reviews tends to be written in the present tense, so only the use of complex verb forms and time adverbials makes them distinct in that regard.

A third dimension (Factor 4 in Biber's taxonomy) involves persuasive or argumentative texts. Here, I add three discourse features (comparative, contrast, and alternative cues) not included in Biber's analysis, but which seem to fit into this category.

Feature	Difference	P-value
Necessity modal (must)	C-Neg > C-Pos	0.021
	C-Neg > I-Pos	0.0015
Possibility modal (could)	C-Neg > C-Pos	.00014
	C-Neg > I-Pos	$5.4 * 10^{-8}$
Prediction modal (would)	C-Neg > I-Pos	0.0096
Conditionals (if)	C-Neg > C-Pos	$2.2 * 10^{-8}$
	C-Neg > I-Pos	$1.6 * 10^{-6}$
Contrasts (but)	C-Pos > I-Pos	$6.5 * 10^{-6}$
	C-Pos > C-Neg	.00014
Comparatives (more)	C-Pos > I-Pos	$7.3 * 10^{-10}$
	C-Pos > C-Neg	$7.0 * 10^{-8}$
Alternatives (or)	C-Neg > C-Pos	$2.6 * 10^{-9}$
	C-Neg > I-Pos	$5.6 * 10^{-5}$

Table 23: Feature P-values for Genre Dimension 3

What is rather striking about these features is that they tend to distinguish either positive comment or negative comment but not both; to make good use of them for separating

[11] Due to an error on my part, only VBN (past participle), and not AUX+VBN (auxiliary + past participle, e.g., *had/is finished*), was included in the final list of features used for training, however VBN appears most often with an auxiliary.

description from comment, it is necessary to include positive and negative comment as two separate categories.

Feature	Difference	P-value
(_NNP	I-Pos > C-Pos	$3.81*10^{-13}$
	I-Pos > C-Neg	$9.75*10^{-11}$
Appreciation	C-Pos > I-Pos	$5.31*10^{-17}$
	C-Neg > I-Pos	$6.04*10^{-21}$
Judgment	C-Pos > I-Pos	.0033
	C-Neg > I-Pos	.0002
Affect	C-Pos > I-Pos	.016
	C-Neg > I-Pos	.007
Text Position	C-Pos > I-Pos	$3.11*10^{-25}$
	C-Pos > C-Neg	$3.55*10^{-7}$
	C-Neg > I-Pos	$8.98*10^{-9}$

Table 24: Feature P-values for Other Genre Features

In Table 24, we examine some other key features that do not fall under Biber's rubric. As expected, proper nouns with parenthesis are strong indicators of description. Nor is it surprising that appreciation is a much more reliable indicator of comment than judgment or affect (which would often appear in character or plot descriptions). The text position numbers indicate that, for positive reviews, description appears earlier in the text, whereas comment tends to appear towards the end; comments from negative texts, which, as we've seen, often lack purely descriptive paragraphs, average out to just beyond the midpoint of the text.

The above is not an exhaustive list of the features used; I choose features which showed significance at the P<0.05 level in at least one relevant comparison, with 44 in all. Most of the rest were POS tags and punctuation, some of which defy easy explanation. For instance, commas are much more frequent in description, a fact which is not intuitive but perhaps can be explained in terms of frequency of relative clauses. In any case, by looking at the P-values for features in the context of previous research I was able to eliminate a large group of irrelevant features and justify a majority of those that were included in our final feature set (see Appendix 3).

4.2.3 Classifier Evaluation

Based on the results presented in the last two sub-sections, I proceeded to built SVM models that classified paragraphs into positive comment, negative comment, description, and formal zones. Rather than eliminate data (as we had previously) to overcome the SVM majority bias with our unreliable dataset, I randomly resampled from the sparser classes to increase the counts. I tried two versions of this: one in which resampling was carried out until there were equal numbers of each class, and one where the counts of each class were increased until they were proportional to their expected rate of appearance in the corpus (based on the sampling results in Table 19). I built 4 classifiers, two of each type on each set of 1000 reviews in the corpus (balanced for polarity), and tested them on our manually annotated dataset; to check for overfitting, I split the test set into two parts based on whether the paragraph had been originally sampled from the first or second 1000 texts. Table 25 gives, for each model/test set combination, the precision, recall, and f-score[12] for classification of description.

Model		TestSet	Precision	Recall	F-Score
Equal Counts	First Half	First	0.67	0.61	0.64
		Second	0.63	0.61	0.62
	Second Half	First	0.67	0.67	0.67
		Second	0.61	0.70	0.65
Prop. Counts	First Half	First	0.73	0.39	0.51
		Second	0.75	0.36	0.48
	Second Half	First	0.73	0.51	0.60
		Second	0.67	0.49	0.57

Table 25: Performance of Semi-Supervised Classifiers on Test Sets

With respect to the correct classification of description (our primary concern), the two resampling methods resulted in two different types of classifiers: one higher recall/lower precision and the other higher precision/lower recall. The section of the corpus trained on seemed to be the next most important factor after the resampling method. There is no evidence of overfitting, in fact the performance of the models built on the second half of the corpus directly suggest the opposite; this is attributable to our compact, targeted feature set. Overall 4-way accuracy hovered near 65% for all the models, with very high accuracy on formal zones (which are overrepresented in our sample), and F-scores of between 0.5 and 0.6 for the two types of comment.

Using these four models I rebuilt two versions of the full-text corpus, one for each resampling method. In this case (despite the lack of overfitting), I made sure there was no testing on training data: the paragraphs from each half of the corpus were tagged with XML tags using the classifier trained on the other half. I then ran these two versions through the SO Calculator with various weights on the spans that had been tagged as descriptive, using the optimal negative weight from Table 1 (1.3) and no weight on formal zones. The results are given in Table 26.

[12] Precision is the number of instances correctly classified as belonging to a class divided by the total number of instances *classified* as belonging to the class, whereas recall is the number correctly classified as belonging to the class divided by the number *actually* belonging to a class; in short, high precision reflects being certain of your classification, high recall reflects correct classification of many members of the class. F-score is a composite of the two:
F = 2*(precision * recall)/ (precision + recall)

Models	Texts	Default Weight (1)			Weight 0.25			Weight 0		
		Pos	Neg	All	Pos	Neg	All	Pos	Neg	All
High Recall	First	73.0	84.0	78.5	78.4	78.2	78.3	80.2	75.8	78.5
	Second	73.2	80.4	76.8	74.8	82.0	78.4	76.2	79.2	77.7
	All	72.8	82.2	77.7	76.6	80.1	78.4	**78.1**	78.0	78.1
High Precision	First	73.0	84.0	78.5	76.0	83.8	79.9	76.0	83.2	79.6
	Second	73.2	80.4	76.8	76.4	78.6	77.5	77.8	78.6	78.9
	All	72.8	82.2	77.7	76.2	81.2	78.7	**76.9**	80.9	78.9

Table 26: Accuracy of SO Calculator with Weights on Semi-Supervised Tagged Paragraphs

The overall trends seem fairly robust; except for a slight drop at the 0 weight using the high recall (low precision) model, which can be explained as the result of complete elimination of mistagged paragraphs (for instance, labeling the only paragraph of a one paragraph text as description, which would always result in misclassification), lowering the weight of description improves performance, and the effect can be viewed across the subsets of the data. In addition, there is steady improvement of positive text classification and a corresponding drop in accuracy for negative texts, which is exactly what we would expect given the data in Table 18; removing description increases positive bias, bringing movie reviews closer to the performance of other product reviews. For both classifiers the change with respect to positive texts was significant (Chi-Squared, p <0.01). The best overall improvement, 1.2%, is not statistically significant but is actually slightly above what we would expect if we assume that description is the sole cause of the 3.3% performance gap from Table 18, given the precision and recall numbers in Table 25; the high-precision classifier is perhaps correctly identifying about a third of all the description in the texts. Note that the high-recall classifier never outperforms the high-precision classifier (despite higher f-scores), which means there is also a significant cost to incorrect tagging.

4.3 Paragraph Classification Using Zone Annotated Texts

4.3.1 Movie Zones Annotation[13]

The previous section made use of a basic annotation schema for movie reviews, a much simplified form of a larger project to identify, at the level of the paragraph, the zones found in movie reviews. This work has expanded on the basic distinctions drawn by Bieler et al. (2007) and Stegert (1993) particularly with respect to functional zones. Instead of a two-way comment/describe distinction, this schema allows for an intermediate option, Describe+Comment, used with paragraphs which contain at least one sentence of both types. In addition, each of the Comment/Describe/Describe+Comment tags has 5 subtags, which are used to identify the target of the comment or description: the options are overall/content, plot,

[13] The English Movie Zones annotation system discussed in this section is primarily the work of Maite Taboada and Manfred Stede. The other annotators for testing inter-annotator agreement are Maite Taboada and Milan Tofiloski.

actors/characters, specific, and general. Specific refers to one aspect of the movie (not plot or characters) which can be manually entered by the annotator, whereas general refers to multiple topics in a single paragraph (which, again, can be listed, e.g., *special effects, cinematography);* this provides built-in tags for standard film review topics, but also some flexibility for handling less common topics. Outside the comment/describe scale, we also include tags like Background (which is directed at past movies or events), Interpretation (which is subjective but not opinionated), and Quotes. All together, the annotation system includes 40 tags, with 22 formal zones and 18 functional zones. Thus far we have annotated 50 published movie reviews (collected from rottentomatoes.com) with this schema. Note that the polarity of the comment is not annotated, so it is not possible to use these reviews to identify positive or negative comment at the paragraph level. The full list of tags is provided in Appendix 4.

The sheer number of tags in our system poses a problem for assessing inter-annotator reliability; the *kappa* statistic, a standard reliability metric (Cohen, 1960), drops as the number of possible categories increase (Sim and Wright, 2005). For our purposes, it is useful to first identify which tags are relevant for sentiment detection, and narrow down or collapse the tags until the number of categories is manageable. After annotating the 50 reviews (which, as usual, maintains an equal balance of positive and negative texts) using the PALinkA discourse annotation software (Orasan, 2003), I converted the PALinka tags into XML tags that the SO Calculator could use for weighting. At this stage I collapsed the formal zones into a single category; the only formal zone that would be directly relevant to the task of sentiment detection is the Rating, and it is *too* relevant; using such information in the calculation would be tantamount to cheating. That said, formal zones do sometimes contain information that could disrupt SO calculation, particularly capitalization has not been preserved, or with certain words in the Audience Restriction zone (e.g., *Rated R for extreme violence*).

Before XML weighting was applied, the baseline accuracy in the 50 review set was 74%, which is a fairly low score, even for movie reviews. This could be because these reviews, unlike those in the Polarity Dataset, are written by professional movie reviewers, and therefore often a great deal more opaque in terms of evaluative language. For example:

(38) It might be an illustration from one of those gift volumes of American history we got as children and left unread. Seeing " Amistad " is a little like looking at pictures without a text to unify them.

The advantage of using professional reviews for annotation is their less haphazard organization, with more paragraphs that are clearly comment or description, and a lot more formal zones (particularly zones like Tagline and Structure). There are some obvious commonalities as well: an interesting one is the fact that even professional reviewers spend less time describing bad movies. In this case, the positive texts included 103 Describe paragraphs or about 2 per review, whereas the same number of negative texts had only 59, or about 1.2 per review.

Working from the 74% baseline performance, I tested different weights on the various functional zones, ultimately boosting accuracy to 82%. The best performing configuration ignored SO valued words in Describe, Background, and Interpretation zones, and reduced the weight of SO words in Describe+Comment zones to one-fourth of their original value (performance drops drastically, back to 74%, if Describe+Comment paragraphs are ignored altogether). Increasing or decreasing the weight of any of the topic-related functional tags (e.g., overall) led to equivalent or often diminished performance. This is surprising, since it seems that Comment-Overall paragraphs would naturally the best indicators of text sentiment. One problem with this reasoning might be that the sentiment expressed in these kinds of paragraphs is frequently couched in metaphor, which an automated system without world knowledge cannot possibly make sense of. Another example from the corpus:

(39) You've Got Mail may not travel the Sammy-Sosa-like distance of the earlier film, but it's over the wall. A homer is a homer.

And, though the appraisal of specific aspects of the movie might not always mirror the overall sentiment, the reviewer will generally choose to discuss aspects of the movie that justify his or her rating. Aspect identification is an interesting problem with has received a fair bit of attention (Hu and Liu, 2004, Titov and McDonald, 2008), however it does not seem directly applicable to the problem of text-level polarity identification, except insofar as the Background tag might be viewed as an example of an aspect of a movie that is unreliable in terms of its contribution to the overall sentiment. For my purposes I have grouped Background and Interpretation under the Describe heading, however, because their counts in the dataset (below 25) are too low to justify separate classes, and like Describe they should generally be disregarded for sentiment analysis. This leaves us with four tag classes: Describe, Comment, Describe+Comment, and Formal.

Prior to full annotation, three annotators (including the author) each annotated the same four texts to test for reliability. Two of the annotators were fairly experienced, having worked together on development and testing of the annotation schema, whereas the third had been brought into the project only recently. Here, I use Fleiss' Kappa (Fleiss, 1971), which extends easily to the case of multiple raters, see Di Eugenio and Glass (2004) for a discussion. Below, I provide the kappa statistic for the four-class case, a three classes case where Describe+Comment has been merged into Comment (as was done for sampled annotation in the previous section), the three class case where Formal zones have been removed from consideration[14], and the two class case with just Comment/Describe. For each case, I give a kappa statistic derived from the ratings of two experienced annotators as well as all three raters.

[14] There was almost perfect agreement on the formal/functional distinction; the one exception was single Tagline which was tagged by the less experienced annotator as a Comment-Overall. For the calculation of kappa with Functional zones, I have eliminated any paragraph which was tagged as a formal zone by any of the annotators.

Classes	2-Rater Kappa	3-Rater Kappa
Describe/Comment/Describe+Comment/Formal	.82	.73
Describe/Comment/Formal	.92	.84
Describe/Comment/Describe+Comment	.68	.54
Describe/Comment	.84	.69

Table 27: Kappa Values for Movie Zones Annotation

In general, the kappa scores were higher when Formal zones are included, since the functional/formal distinction is very clear; the use of Describe+Comment, on the other hand, made reliable annotation more difficult. There is no universal standard to evaluate Kappa scores, however κ > .67 has been used as a standard for reaching conclusions in Computational Linguistics since Krippendorf (1980). Only one of the kappa values presented in Table 27 falls below that standard, and the κ > .8 seen with the Describe/Comment/Formal distinction would allow for more definite conclusions. Under other, more liberal standards (Di Eugenio and Glass, 2004, Rietveld and van Hout, 1993), all the values indicate at least moderate agreement.

After the categories were collapsed, the 50 annotated texts contained 332 paragraphs tagged Formal, 171 paragraphs tagged Comment, 158 paragraphs tagged Describe, and 156 paragraphs tagged Describe+Comment.

4.3.2 Evaluation of Supervised Classifiers

Having established the annotation scheme, we now turn to using the annotated texts to identify zones in new texts (again, Movies, i.e., the Polarity Dataset) for the purpose of sentiment analysis. We use two basic feature sets: the 44 significant genre features based on Biber 1988 and discussed in the preceding section, and, following Bieler et al. (2007), a 5-gram classifier, including binary features indicating the presence of single words and sets of 2, 3, 4, and 5 consecutive words which appeared at least 4 times in the dataset (this kept the number of features to about 8000); note that there was only slight improvement in cross-validated classification performance past 3-grams, since, except for names and titles, very few 4- and 5-grams repeated in our dataset. Bieler et al. used an SVM classifier, however preliminary testing found that a Bayes Naïve classifier seemed to work best for N-gram features on the small dataset, giving about 80% accuracy on the 3-way classification of Describe/Comment/Formal (10-fold cross-validation) as compared to 76% for SVMs, as well as higher precision for the identification of Describe. With the genre features, however, the SVM classifier was again preferred. Our precision and accuracy for just the functional zones were significantly lower than Bieler et al., which could be attributed to any of a number of factors, including our smaller dataset and the fact that Bieler et al. were working in a different language (German). Table 28 is similar to Table 25, providing information about the identification of description using different models.

Model	Training set, 10-fold crossvalidation			Test Set Sampled from Movies (see Section 2.1)		
	Precision	Recall	F-score	Precision	Recall	F-score
5-gram Bayes 2-way	0.63	0.57	0.60	0.69	0.56	0.62
Genre SVM 2-way	0.71	0.57	0.63	0.69	0.69	0.69
5-gram Bayes 3-way	0.67	0.55	0.60	0.71	0.54	0.61
Genre SVM 3-way	0.64	0.43	0.51	0.66	0.54	0.59
5-gram Bayes 4-way	0.71	0.56	0.62	N/A	N/A	N/A
Genre SVM 4-way	0.55	0.60	0.57	N/A	N/A	N/A

Table 28: Accuracy for Describe Class using Various Models/Test Sets

Note that for the 4-way classifier, the Describe+Comment accuaracy is also very relevant: for the 5-gram classifier, the precision/recall/accuracy was 0.44/0.61/0.50, and for the genre feature classifier it was 0.40/0.32/0.36. In general, the scores are quite comparable to the numbers for the high recall classifier trained using data automatically selected by the SO Calculator, however there was some variation, for instance the good performance of the 2-way discourse classifier, and the poor performance of the 3-way discourse feature classifier; the 5-gram classifier, on the other hand, was fairly consistent regardless of the number of classes. I also tested a merged model, with both 5-gram and discourse features as well as a meta-classifier that used the output of both classifiers; neither approach seemed to improve performance in either identification of zones or text polarity detection.

For each model, a new version of the Movies corpus was built with all paragraphs tagged using the model. Table 29 gives the accuracy of the SO calculator with various weights on Describe. For the 3- and 4-way models, no weight in put on Formal, and for the 4-way models, the weight on Describe+Comment is 0.25 more than Describe (following the results from optimization of training data performance). For simplicity I have omitted information about positive and negative accuracy, since the patterns are mostly analogous to what we saw in Table 26. Recall that baseline performance without weights is 77.7%.

Classifier	Weight 0.75	Weight 0.5	Weight 0.25	Weight 0
5-gram Bayes 2-way	77.80	78.50	78.50	78.00
Genre SVM 2-way	78.10	78.95	79.00	78.15
5-gram Bayes 3-way	77.80	78.35	78.30	77.90
Genre SVM 3-way	76.25	76.85	76.80	76.55
5-gram Bayes 4-way	77.40	77.75	77.90	76.75
Genre SVM 4-way	75.40	75.30	75.00	72.15

Table 29: Accuracy of SO Calculator with Diff. Classifiers and Weights on Describe Paragraphs

Only one of the models in Table 29 beats out the best accuracy (78.9%) seen by the high precision classifier in the previous section, and in many of the other cases these classifiers actually perform below baseline, a fact which is probably attributable to the poor precision on Comment and/or Describe+Comment, which leads to significant amounts of comment being disregarded; there does seem to be a fair bit of correlation between the numbers in Tables 28 and 29, and the best performing classifier in both cases is the 2-way Comment/Describe SVM, built using the 44 discourse features. We can confirm that this classifier is doing its job (following Pang and Lee, 2004) by reversing the weights, giving Describe a 1 and Comment a zero; under these conditions the accuracy drops to 53.5%, just above chance.

There is another way to integrate the information provided by Describe+Comment without resorting to a multi-class situation, which, based on what we have just seen, has significant drawbacks, particularly for the genre feature classifier. Instead of conceptualizing the tags as three separate classes, we can view them as points in a continuous spectrum, and use another machine learning algorithm, linear regression (Witten and Frank, 2005). Linear regression is based on the same basic mathematical model as an SVM (with weights on various features), however the output is a real number rather than a class. To train this kind of model, all Describe paragraphs are assigned a Comment value of 0, all Comment paragraphs assigned a value of 1, and the Describe+Comment paragraphs assigned a value in-between (or ignored entirely); the paragraphs in the texts will then each be assigned a real-number Comment value based on the best-fit line for this data. It turns out that building a linear regression model using the 5-grams as features is not feasible, even a small fraction of those 8000 features (542 features that appeared at least 15 times in the dataset) took a long time to build, and resulted in a completely useless constant value classifier; the best N-gram classifier tested had a correlation co-efficient of about .3 (1 is perfect correlation), as compared to more than .56 for the best genre feature model; below, for simplicity, I only consider genre-based models. In any case, the Comment value assigned by the classifier is used as the weight in SO Calc, taking advantage of the numeric XML tag feature. Results for various options are given in Table 30. When the test sample was used, I built two separate classifiers for each half of the corpus so there was no testing on data which had been used for training,

Model	Accuracy
Describe+Comment = 0.25	78.70
Describe+Comment = 0.5	79.05
Describe+Comment = .75	79.00
No Describe+Comment	79.15
No Describe+Comment, add sampled test set to training	78.85

Table 30: Accuracy of SO Calculator with Linear Regression Tagging

Linear regression provides the best performance yet when only Comment and Describe are used for training. This does not, however, mean that the Describe+Comment class is superfluous in our annotation schema: consider the drop in performance when the sampled test set is added to the training set, a fact which is best explained by the presence of description mixed in with the comment (recall that I originally classified sample paragraphs that had any comment whatsoever as comment, including many paragraphs that would be Describe+Comment under our full schema). Describe+Comment is useful, then, as a buffer zone, to ensure that Comment paragraphs do not have (much) description and Description paragraphs do not have (much) comment, allowing for more accurate training.

Clearly, the classifiers described here could likely benefit from a much larger training set (including more informal reviews), more testing with various features and types of classifiers, and a comparison with sentence-based approaches. The preliminary results presented here are, however, fairly promising, particularly when linear regression is used with carefully selected genre features from annotated texts (additional testing showed that linear regression worked less well with unreliable data). Surprisingly, the best linear regression model makes use of only 11 features: text position and the frequencies of question marks, commas, nouns, proper nouns with parentheses, 3rd person pronouns, appreciation words, comparatives, *it*, adjectives, and adverbs.

4.4 Text-Level Review Subgenre Classification

In the previous two sections, I simply assumed the optimal negative weight as a starting point for genre-based improvements; however, as we saw in Table 18, changing the negative weight from our default weight of 1.5 in either direction leads to worse performance because the optimal weights for movie reviews and camera/printer reviews are quite different. In order to justify this assumption, we need to show that these kinds of reviews can be distinguished accurately. To this end, I built a simple classifier using the 50 annotated movie reviews and 50 camera/printer reviews from the same source as the Camera corpus (though obviously involving different texts). For this task, a unigram classifier seemed most appropriate, since the appearance of individual nouns and verbs would be the most obvious way to distinguish between these texts. Indeed, 10-fold cross-validation using an SVM classifier trained on unigram features (again, those which appeared at least 4 times) yielded 100% accuracy.

I combined our three main corpora into a single corpus, mixing the various kinds of reviews so that their genre could be “rediscovered” by the classifier. Prior to this, all the texts had been tagged using the best performing paragraph classifier from the preceding section. After the texts were classified, they were automatically placed in two separate directories and the SO Calculator was run using the appropriate configuration file: for the texts classified as camera reviews, the weighting option was disabled, and the negative weight was set to 1.7; for texts classified as movie reviews, the weighting option was enabled, and the negative weight was set to 1.3.

First, Table 31 shows how the texts of the various corpora (including Epinions sub-corpora) were classified by the text genre classifier.

Corpus	**% Movie**	**% Camera**
Camera	0	100
Movie	97.25	2.75
Epinions	36.25	63.75
Epinions:Movie	82	18
Epinions:Books	62	38
Epinions:Music	74	26
Epinions:Hotels	28	72
Epinions:Cars	34	66
Epinions:Phones	0	100
Epinions:Computers	10	90
Epinions:Cookware	4	96

Table 31: Percentage of Each Corpus Classified as Each Review Genre

The classifier was able to distinguish not only movie and camera/printer reviews (which it did quite well, not a single camera review was tagged as a movie review), but also two general review subgenres: cultural products (movies, books, and music) and physical products (hotels, cars, phones, computers, and cookware).

Table 32 compares SO Calculator accuracy between performance with default configuration and performance with genre-specific configurations.

Corpus	**Configuration**	
	Default	**Genre-specific**
Camera	80.3	81.1
Movie	76.0	**79.1**
Epinions	80.25	80
Epinions:Movie	84	84
Epinions:Books	72	80
Epinions:Music	82	78
Epinions:Hotels	72	74
Epinions:Cars	90	90
Epinions:Phones	80	76
Epinions:Computers	94	86
Epinions:Cookware	68	72

Table 32: SO Calc Accuracy with Genre-Specific Configurations

The improvement to the performance of the Movies is significant at the $p < 0.01$ level. Although the overall effect of our genre classification efforts on the Epinions corpus is slightly negative, the effect on individual subcorpora is telling: Books jumped 8 percent, probably because it benefits from paragraph weights that eliminate plot description; Music, on the other hand, has a lot of movie-tagged texts but is unlikely to contain that kind of description, so its performance drops. In general, the mixed performance of this mixed corpus indicates that it is likely an oversimplification to suppose there are only two review subgenres, or that just two possible SO Calc configurations would be enough.

Finally, we tested the effect of description weighting in technical texts by turning XML weighting on for the Camera corpus. The accuracy dropped slightly, to 80.8%, which suggests that our divide and conquer approach is a good one; it is not appropriate to look for and disregard apparent plot/character description in camera reviews which have none, though there might be parts (stages) of the text in this review subgenre that can be identified and similarly ignored or discounted. The other advantage of identifying multiple subgenres would be the option to use subgenre-specific dictionaries (a feature the SO Calculator supports): for instance, the word *slow* might be taken as a negative word when encountered in technology or service reviews but viewed as purely descriptive in the context of a music review.

In this chapter, I have shown how genre classification techniques can be applied to improve SO Calculator performance. From the perspective of sentiment analysis, there is clear benefit to identifying the genre of a review, and the genres of paragraphs contained within.

Chapter 5: Cross-Linguistic Sentiment Analysis

Much of the work in sentiment analysis has been focused on English, but this is rapidly beginning to change. In this chapter, I discuss the application of sentiment analysis techniques to other languages, with a focus on sentiment-relevant linguistic differences as well as adaptation of our SO calculator. In the first section I provide a comprehensive review of relevant literature on cross-linguistic automated sentiment analysis, which includes both semantic and machine learning approaches in a growing number of languages. Section 2 of this chapter is concerned with a Spanish version of the SO calculator, and the question of whether machine translation is a good alternative to building language-specific resources. In Section 3, I put aside computational concerns and look at Chinese from a linguistic perspective, highlighting some features of the language which seem relevant to sentiment and counting the appearance of these features in a small review corpus.

5.1 Previous Research

Chinese was one of the first languages after English to receive attention from opinion mining researchers, and it is fairly safe to say that there is more work in Chinese than any other language besides English, including semantic models (Hu et al., 2005, Ku et al., 2005, Qiu et al., 2007, Wu et al., 2007, Yao et al., 2006, Ye et al., 2006), machine learning approaches (Li and Sun, 2007, Tan and Zhang, 2008, Wang et al., 2007), as well as direct comparisons of the two (Ye et al., 2005). There has also been a great deal of work in Japanese (Hiroshi et al., 2004, Kaji and Kitsuregawa, 2007, Takamura et al., 2005, Wang and Araki, 2008), and English, Chinese, and Japanese are the three languages included in the Opinion Analysis Task at the annual NTCIR (Seki et al., 2007, Seki et al., 2008); this task involves detection of opinion, opinion polarity, opinion holder, and opinion target at the sentential level. Other languages that have received particular attention include Korean (Cho and Lee, 2006), Romanian (Mihalcea et al., 2007), French (Bestgen, 2008) and Arabic (Abbasi et al., 2008).

For semantic models, dictionary building is always a fundamental problem, especially since other languages often initially lack the resources available in English (WordNet, for instance). Cho and Lee (2006), who are interested in a very detailed spectrum of sentiment, built their dictionary manually, assigning each Korean word an emotional vector (averaged across multiple judges) with dimensions such as *sadness, excitement* and *surprise*. Researchers in Chinese (Lu et al., 2008) have also made use of existing manually-created resources such as dictionaries of positive and negative terms (Shi and Zhu, 2006, Yang and Zhu, 2006). One interesting question is to what extent existing (manual) resources in English are useful for building dictionaries in new languages. Yao et al. (2006), for instance, report good results from a system that determines the polarity of Chinese words based on their English version in a Chinese-English lexicon. Mihalcea et al. (2007), however, conclude that, at least for the task of sentence subjectivity detection, translating words to Romanian from an existing English subjectivity dictionary is not a viable approach, since a great deal of the subjectivity (sentiment) is lost in translation.

One language-specific approach that deserves special attention is Ku et al. 2005. They use the fact that multi-character Chinese words are built out of a (relatively) small set of Chinese characters which generally have their own independent meanings. Starting with a seed list of positive and negative words, they use information about frequency of character appearance to calculate an SO value for each character. Then they average the SO values of consistent characters to compute an SO value for novel words. They note that this method seems to capture not only basic polarity, but also intensity. A large set of training words is required to bootstrap the system, however.

Methods for fully-automated dictionary building have been adapted and expanded in other languages. Wang and Akari (2008), for instance, adapt the Turney's SO-PMI (hit count) algorithm to Japanese, while Kaji and Kitsuregawa (2007) use simple patterning matching techniques (structural cues, e.g., pros and cons) to extract huge amounts of positive and negative Japanese data from the web, using two different metrics (including PMI and Chi-square-based values) to extract polarity information from these texts. The latter group reports that their method outperforms hit-count-based PMI, noting that they were able to identify the polarity of colloquial words that would not normally appear in bilingual dictionaries.

In Chinese, there have been at least two comparisons of machine learning algorithms for sentiment analysis. Tan and Zhang (2007) found that, as in English (Pang et al., 2002), SVMs provided the best performance in a text identification tasks, while Li and Sun (2007) also report good performance using SVMs, but suggest that a Naïve Bayes classifier might be better depending on the features chosen. As with dictionary building, the special compositionality of Chinese words allows for interesting choices with respect to features; both Zagibalov (2008) and Li and Sun (2007) both found that character-based features were useful when added on top of more traditional word-based features. Tan and Zhang focus on feature selection algorithms in Chinese, deciding that feature selection based on information gain is the best approach.

Abbasi et al., (2008) offers a direct comparison between sentiment analysis feature selection in English and Arabic. First of all, the starting feature sets in the two languages are different, due to the basic properties of the language, e.g., Arabic is morphologically rich, and so roots are used in addition to the unigrams popular in English, and special morphological changes (such as elongation to emphasize key words) are included as features. Interestingly, the usefulness of certain features in the exact same domain (posts on extremist forums) varied considerably between languages, in general the feature selection algorithm chose many more syntactic and stylistic features in English than in Arabic, only roots, function words, and individual letter features were more useful feature in Arabic. Nevertheless, SVM classifier performance in the two languages was quite comparable.

Bautin et al. (2008) suggests a different method for tackling the problem of doing sentiment analysis in many different languages: use a single existing sentiment analyzer for English, translating the texts using state-of-the-art machine translation technology. Using the Lydia system (Lloyd et al., 2005), their goal was to track attitudes toward newsmakers as reflected in various online media in countries around the world. For their purposes, the particular translation system did not seem to be important, and they were able to note clear similarities and differences in attitude across various languages, noting, for instance, that certain languages seem to have more positive or negative bias (Korean was the most positive, Italian the most negative).

Wan (2008) also makes use of machine translation, but for the task of polarity detection at the level of text. Given that the resources for sentiment analysis in Chinese are limited, he translates his corpus into English using several different translation systems (he notes that Google seems to be the best) and combines the results from word-counting of the various English versions as well as a basic Chinese system; a final SO score is calculated from the weighted average of the various individual scores. This improves performance significantly as compared to the Chinese baseline, and in fact the SO information coming directly from Chinese is barely used in the final calculation. The direct translation of English dictionaries was also tried (to improve the performance of the Chinese calculator), but did not seem to help. It is not clear, however, exactly why the Chinese system performed so poorly compared to the English system, and in general the study, though suggestive, is far from conclusive with respect to the long term potential of machine translation as compared to language-specific resources and machine learning algorithms. In the next section, we compare various alternatives in a new language, Spanish.

5.2 Spanish Text Sentiment Analysis[15]

5.2.1 The Spanish SO Calculator

Our primary approach to sentiment analysis in Spanish is the use of a modified version of the English SO calculator, including the creation of Spanish dictionaries.

5.2.1.1 Adapting the Calculator

Compared to English, Spanish is a highly inflected language, with gender and plural markers on nouns, as well as a rich system of verbal inflection (a total of 45 possible verb forms). In the English version of the SO Calculator, the only external software we made use of was the Brill tagger (Brill, 1992); lemmatization of noun and verbs was simple enough to be carried out during the calculation. For Spanish, we used a high-accuracy statistical tagger, the SVMTool (Giménez & Màrquez, 2004), and we adapted a 500,000+ word lemma dictionary included in the

[15] This section is taken from a paper (Cross Linguistic Sentiment Analysis: From English to Spanish) which has two other authors, Milan Tofiloski and Maite Taboada (I am the head author). Much of Section 2.2.1 was originally written by Maite Taboada, and a significant portion of the corpus and dictionary building are due to the efforts of my co-authors.

FreeLing software package[16], which we used to both lemmatize the words and also add additional detail to the basic verb tags assigned by SVMTool; i.e., each verb is lemmatized to its infinitive form but tagged with information about its original tense and mood. We found that some sentiment-relevant nouns and adjectives were not being lemmatized properly (they were not in the lemma dictionary), so we also implemented a second layer of lemmatization within the calculator.

Most of the Python code written for the English version of the calculator could be reused with little or no modification. With regards to detecting negation, intensification, and modifier blocking, it was necessary to take into account the fact that in Spanish adjectives appear both before and (more commonly) after the noun. In addition, some adjectival intensification in Spanish is accomplished using suffixes, for instance *–ísimo*, which we treated as a superlative similar to English *–est*; there are also other morphological markers that can express sentiment (e.g., *–ito*, used for diminutives), but they are rarer and harder to quantify. For our purposes the most interesting difference was the fact that verb forms in Spanish give irrealis information not always available in English. In particular, the conditional tense and the imperative and subjunctive moods often serve to indicate that the situation being referred to is not in fact the case (if it were, the indicative would be used). A good example of this is certain relative clauses:

(40) Buscamos un puesto que sea interesante
Looking -for a job that is-SUBJ interesting
I'm looking for a job that is interesting.

Here, the interesting job is entirely hypothetical, a fact that is reflected directly by the use of the subjunctive in the Spanish but available only through consideration of verb semantics in the English translation. Thus, in Spanish we used a mixture of word and inflection-based irrealis blocking, using the same words as the English version whenever possible.

One other interesting feature of Spanish which we have not yet integrated into our model is the semantics associated with the placement of adjectives. Many adjectives in Spanish can be placed either before or after the noun, however the interpretation is often radically different (González and Farrell, 2001). Adjectives that appear before the noun tend to be interpreted in as a subjective evaluation, whereas adjectives appearing after the noun are often interpreted as being descriptive (especially when there are multiple adjectives). An example would be the adjective *grande*, which means 'great,' 'famous' when it appears before the noun, but simply 'big' or 'tall' when it appears afterwards. It is unlikely that integrating information such as this would lead to any great performance gains, but it is one potential avenue for gradual improvement of the model, perhaps as part of a more general attempt at word sense disambiguation.

[16] http://garraf.epsevg.upc.es/freeling/

5.2.1.2 Building New Dictionaries

We built new Spanish dictionaries, analogous to the English ones, including dictionaries for adjectives, nouns, verbs, adverbs, and intensifiers. For intensifiers, given the fact that they are closed-class and highly idiosyncratic, we simply created a new list of 157 words and expressions based on the English list. For the open-class dictionaries, we tested three different methods of dictionary-building; we compare their performance on the Spanish corpus in Section 2.3.

The first set of dictionaries started with the English dictionaries for each part of speech, which we translated automatically into Spanish, preserving the semantic orientation value for each word. For the automatic translation we used, in turn, two different methods. The first was an online bilingual dictionary, from the site www.spanishdict.com . There, we extracted the first definition under the appropriate syntactic category, ignoring any cases where either the English or the Spanish were multi-word expressions. The second automatic translation method involved simply plugging our English dictionaries into the Google translator and parsing the results (again excluding multiword expressions). Note that the latter method may result in the wrong part-of-speech assignation or incorrect (non-lemma) form. Table 33 shows the size of dictionaries for each method, by part of speech.

For the second method of dictionary creation, we took the lists from spanishdict.com, and manually fixed entries that were obviously wrong. This involved mostly removing words that were in the wrong dictionary for their part of speech, but also changing some of the values (less than 10% for each dictionary). This hand-correction took a native speaker of Spanish about two hours to complete.

Finally, the third method consisted in creating all dictionaries from scratch. Our source corpora created for this project consists of reviews extracted from the ciao.es (Ciao) consumer review website. Following the basic format of the Epinions corpus, we collected 400 reviews from the domains of hotels, movies, music, phones, washing machines, books, cars, and computers. Each category contained 50 reviews: 25 positive (1 or 2 stars), and 25 negative (4 or 5 stars). Whenever possible, exactly two reviews, one positive and one negative, were taken for any particular product, so that the machine learning classifier would not be able to use names as sentiment clues.

We first tagged the Spanish corpus (the Ciao, i.e., the development corpus), and then extracted all adjectives, nouns, adverbs and verbs. This resulted in large lists for each category (for instance, the noun dictionary had over 10,000 entries). We manually pruned the lists, removing words that did not convey sentiment, but also misspelled words, words in the wrong part of speech, and inflected words. Finally, semantic orientation values were assigned for each. This pruning and assignation process took about 12 hours (performed, again, by a native speaker of Spanish). For various reasons, we decided against a full committee review of the Spanish dictionaries for the time being.

Another type of dictionary that we tested was a merging of the dictionaries created using the second and third methods, i.e., the automatically (but hand-fixed) dictionaries and the ones created from scratch. We created two versions of these dictionaries, depending on whether the value from the Fixed Spanish-dict.com value or Ciao dictionary value was used. The size of these combined dictionaries is comparable to the size of our original English dictionaries.

Source	**Size of Dictionary by POS**			
	Adjective	**Noun**	**Verb**	**Adverb**
Spanishdict.com	1160	879	500	422
Google translated	1331	752	583	673
Spanish-dict fixed	1150	871	500	416
Ciao corpus	1465	689	379	168
Ciao/fixed combined	2049	1324	739	548

Table 33: Size of the Spanish Dictionaries

We performed a comparison of fully automated and fully manual methods, comparing the unedited spanishdict.com dictionaries and the ones created by hand from scratch. First, we calculated the percentage of words in common, as a percentage of the size for the larger of the two sets (the spanishdict.com dictionaries). The commonalities ranged from roughly 20% of the words for nouns to 41% for adjectives (i.e., 41%, or 480 of the hand-ranked adjectives were also found in the automatic dictionary). We also compared the values assigned to each word: The variance of the error ranged from 1.001 (verbs) to 1.518 (adjectives). In summary, we were more likely to include the same adjectives in both dictionaries, however the SO value for those adjectives were the most prone to variation. A visual inspection of the two types of dictionaries reveals that automatically translated dictionaries tend to include more formal words, whereas the one created by hand includes many more informal and slang words, since those words come directly from the reviews. It is also worth pointing out that, for an informal or slang English word appearing out of context, the online dictionary often seemed to produce a more formal equivalent in Spanish.

5.2.2 Alternative Approaches

5.2.2.1 Corpus Translation

For translation of our corpora, we used Google's web-based translation system. Google Translate[17] uses phrase-based statistical machine translation; however, detailed information about its workings is unavailable, since it is proprietary technology. We used only one translator, but see Bautin et al. (2008) for a discussion on using different Spanish translating systems, and Wan (2008) for a comparison of Chinese machine translators; the latter found that Google gave the best performance, which is consistent with our preliminary testing.

[17] http://translate.google.com

5.2.2.2 Machine Learning

A popular approach to sentiment analysis has been the automatic training of a text classifier. Cross-linguistic sentiment detection seems particularly amenable to machine learning, since classifiers can be easily trained in any language. Following Pang et al. (2002), we used an SVM classifiers, built with the sequential minimal optimization algorithm included in the WEKA software suite (Witten & Frank, 2005), with a linear kernel and testing done with 10-fold cross-validation. We trained using unigram features that appeared at least four times in the dataset. To test the efficacy of the WEKA classifiers, we first trained a classifier on the full 2000 text Polarity Dataset, comparing the cross-validated results with the baseline for SVM unigram classifiers on this dataset (before other improvements) given in Pang and Lee (2004). The difference (about 1%) was not statistically significant. It is worth noting that more recent work in SVM-based sentiment analysis has shown significant improvement on this baseline (e.g., Whitelaw et al. 2005, Abbas et al. 2008), however relevant resources are not presently available in Spanish.

5.2.3 Evaluation

We built two additional 400 text corpora, one English and one Spanish, with the same basic constituency as the Epinions and Ciao Corpus discussed earlier. The English corpus (Epinions 2) is also from Epinions (we made certain there was no repeat texts), while the Spanish corpus (Dooyoo) was from a different website, dooyoo.es. All four corpora were translated using the appropriate Google translator, and for each version the accuracy identifying the polarity of reviews for all possible dictionaries and methods was tested. Note that when the corpus and the dictionary are the same language, the original version of the corpus is used, and when the corpus and the dictionary are of different languages, the translated version is used. Recall that the Subjective dictionary (evaluated in Chapter 3) is based on the subjectivity cues of Wilson et al. (2005).The results are given in Table 34.

There are a number of clear patterns in Table 34. First, for the original Spanish versions, the translated Spanish dictionaries, taken together, do poorly compared to the versions of the dictionaries derived from actual Spanish texts; this is significant at the $p < 0.05$ level (all significance results are derived from chi-square tests) for all possible dictionary combinations. For Spanish, including words from translated dictionaries has little or no benefit. The opposite is true for Spanish translations of English texts, where the Ciao dictionary performance is low, and performance improves dramatically with the addition of translated (though manually fixed) resources; in the case of the Epinions 2 corpus, this improvement is significant ($p < 0.05$). We attribute this to the fact that translated texts and translated dictionaries “speak the same language” to a certain extent; translated English corpora are unlikely to contain colloquial Spanish such as is found in the Ciao dictionary, and are more likely to contain kind of formal language we saw in our translated dictionaries.

		Corpus				
Method		English		Spanish		
Calculator	Dictionary	Epinions	Epinions2	Ciao	Dooyoo	Overall
English	Subjective	67.75	69.00	63.50	66.75	66.75
English	Main SO Calc	**80.25**	**79.75**	72.50	**73.50**	76.50
Spanish	Google-translated	66.00	68.50	66.75	66.50	66.50
Spanish	Spanishdict.com	68.75	68.00	67.25	67.25	67.94
Spanish	Fixed Spanishdict.com	69.25	69.75	68.25	68.00	68.81
Spanish	Ciao	66.00	67.50	**74.50**	72.00	70.00
Spanish	Ciao + Fixed, Ciao value preferred	68.75	**72.50**	74.25	72.25	**71.93**
Spanish	Ciao + Fixed, Fixed value preferred	69.50	68.75	73.50	70.75	70.87
SVM, English versions		76.50	71.50	72.00	64.75	71.25
SVM, Spanish versions		71.50	68.75	72.25	69.75	70.56

Table 34: Accuracy of Polarity Detection for Various Corpora and Methods

Turning now to our main comparison, the SVM classifiers show the worse performance overall, however only the difference seen in the Epinions 2 corpus is significant (at the $p < 0.01$ level). The relatively poor performance of the SVM classifier in this case can be attributed to the small size of the training set and the heterogeneity of the corpora; SVM classifiers have been shown to have poor cross-domain performance in text sentiment tasks (Aue and Gamon, 2005), a problem that can be remedied somewhat by integrating a lexicon-based system (Andreevskaia and Bergler, 2008).

The numbers in Table 34 do not indicate a clear winner with respect to the performance of the Spanish SO Calculator as compared to the English SO calculator with translated texts, though it is clear that translating English texts into Spanish for analysis is, at present, a very bad approach ($p < 0.01$). Moreover, the totals for all corpora for each method suggest that the Spanish SO Calculator is performing well below the English SO Calculator ($p < 0.01$).

In order to look at the broader trends in the effects of translation, it is necessary to recombine the results into a different format. In Table 35, Original refers to all the 1600 original versions and Translated refers to all 1600 translated versions. For the SO Calculation, we use the best performing dictionary in the relevant language.

Method	Texts	Accuracy
SO Calculation	Original	76.62
	Translated	71.81
SVM	Original	72.56
	Translated	69.25

Table 35: Accuracy for Translated/Original Corpora

Table 35 shows a general deficit for translated texts; for SO Calculation, this is significant at the $p < 0.01$ level. The fact that it is also visible in SVMs (which are not subject to dictionary biases) suggests that it is a general phenomenon. One potential criticism here is our use of corpora whose words were the basis for our dictionary, unfairly providing two of the four original corpora with high coverage which would not pass to the translations. Indeed, there is some evidence in Table 34 to suggest that these high coverage corpora do outperform their low coverage counterparts to some degree in relevant dictionaries (compared with the Subjective dictionary, for instance); in general, though, there were no significant differences among same-language corpora tested using the same dictionary. Note also that using high-coverage corpora is not analogous to testing and training on the same corpora, since words are rated for SO independently of the texts in which they appear. Instead, these high-coverage corpora provide a view of overall performance in a future stage of development when general coverage is higher than at present. In this case, the Ciao dictionary in the Ciao corpus provided the best of all Spanish results.

Finally, we looked at the effect of the various features of the SO calculator (see discussion in Chapter 3) by disabling them. We tested both original and translated texts in three dictionaries: the main SO Calc dictionary, the Ciao dictionary, and the (original) Spanishdict.com dictionary translated from the main SO Calc dictionary.

Corpus+Dictionary	Baseline	No Neg	No Int	No Irrealis Blocking	No Negative Weighting	No Repetition Weighting
Epinions + Main SO Calc	80.25	75.75	78.75	78.75	71.75	81.25
Epinions2 + Main SO Calc	79.75	76.50	78.25	78.25	72.25	78.75
Ciao (t) + Main SO Calc	72.50	70.00	72.25	72.25	68.25	72.50
Dooyoo (t) + Main SO Calc	73.50	71.25	72.50	74.00	70.25	72.25
Epinions (t) + Ciao	66.00	65.25	66.50	64.25	60.00	66.00
Epinions2 (t) + Ciao	67.50	68.75	67.75	68.50	61.00	67.00
Ciao + Ciao	74.50	72.75	73.50	71.50	67.75	74.25
Dooyoo + Ciao	72.00	71.50	71.50	70.25	66.00	71.25
Epinions (t) + Spanishdict	68.75	68.00	67.25	66.00	59.50	69.25
Epinions2 (t) + Spanishdict	68.00	66.75	68.75	67.50	58.75	68.25
Ciao + Spanishdict.com	67.25	65.25	67.25	66.75	59.25	68.50
Dooyoo + Spanishdict.com	67.25	65.25	66.75	67.00	63.00	67.50

Table 36: Effect of SO Calculator Features on Accuracy in Various Corpora/Dictionaries[18]

With the exception of repetition weighting in the Epinions corpus, the features of the SO calculator always have a positive effect on performance when neither the corpus nor the dictionary has been translated; though there is obvious variation in the effectiveness of features

[18] Neg = negation, Int = intensification, (t) = translated.

across languages, there is only moderate within-language variation. However, when translated corpora or dictionaries are used, only the strongly positive effect of negative weighting is consistent; the other features show erratic behavior, with each feature having a negative effect on performance in at least one corpus. This is not altogether surprising since features that modify the SO value of words depend crucially on those initial SO values being reliable. Looking at the non-translated results, irrealis blocking is indeed more useful in Spanish, though negation and intensification are less so; we may need to further adapt these latter futures so that they can better handle Spanish syntax (where word order is less rigid).

5.2.4 Discussion

For calculation of semantic orientation using lexicons, translation of any kind seems to come with a price, even between closely related languages like English and Spanish. Our Spanish SO calculator is clearly inferior to our English SO Calculator, probably the result of a number of factors, including a small, preliminary dictionary, and a need for additional adaption to a new language. Translating our English dictionary also seems to result in significant semantic loss, at least for original Spanish texts. Although performance of Spanish texts translated into English is comparable to native SO Calculator performance, the overall accuracy of translated texts in both English and Spanish suggests that there is 3-5% performance cost for any (automated) translation. This, together with the fact that translation seems to have a disruptive effect on previously reliable improvements as well as the relatively small time investment required to get the Spanish SO Calculator to this stage, lead us to conclude that there is value in pursuing the development of language-specific resources for sentiment analysis, notwithstanding new breakthroughs in machine translation.

5.3 A Sentimental Investigation of Chinese

In this section, we turn to another language, Chinese[19]. From an implementation standpoint, Chinese text has a few key orthographic features that result in initial difficulties: for instance, the lack of spaces between lexical words and lack of capitalization to distinguish proper nouns (properties it shares with other written Asian languages), which both require special attention when doing automated sentiment analysis (Wu et al., 2007). However, we shall put aside these kinds of concerns, and focus on linguistic differences between English and Chinese that could potentially be integrated into a deep semantic model. After I have identified a few interesting candidates, I will present the results of a small corpus study of online reviews where the frequency of these features is observed.

[19] Since the syntax and semantics of modern written Chinese are effectively standardized, I use the general term Chinese rather than Mandarin or Cantonese, which refer to types of spoken Chinese. That said, I use the Mandarin *pinyin* Romanization and the simplified character set of mainland China, and there are aspects of this discussion which might not apply to texts written in a colloquial Cantonese-style.

5.3.1 Potentially Sentiment-Relevant Features

From a certain perspective, Chinese and English have more in common than English and Spanish. Both Chinese and English are fundamentally SVO, with fairly rigid word order and relatively little inflection. Neither has the rich system of verbal inflection that we saw in Spanish, and both rely heavily on light verbs and modals. Indeed, much of the content of Chapter 3 can be applied directly to Chinese with only superficial modification. That said, there are some intriguing differences that might be relevant to sentiment analysis. In the discussion that follows, I use information and examples from Li and Thompson (1981), Sun (2006), Wei (1997) and my own (fluent but non-native) intuitions, which I confirmed with native speakers.

First, let us consider adjectives and adjectival phrases, which of course are fundamental to sentiment analysis. Chinese has a number of ways to associate an adjective with a noun as a proposition, the choice of which depends on the nature of the adjective and the intent being expressed. Perhaps the most obvious difference compared to English is that the copula cannot be used directly with an adjective.

(41) *Wo shi pang
I am fat

Instead, a speaker may use any one of four other alternatives.

(42) a. Wo hen pang
I very fat
b. Wo shi pang de
I am fat NOM[20]
c. Wo pang
I fat
d. Wo pang le
I fat CRS
I am fat.

The nominalizer *de* is used generally to change an adjective or verb phrase into a noun, and the sentence final particle *le* has a number of uses, here it could indicate a change of state or a flaunting of expectation (we will discuss sentence final particles later in this section). (42a) is probably the most common way to express this idea, even though it seems to require adding additional meaning (an intensification). Li and Thompson argue, however, that *hen* is semantically bleached, and should not be treated as an intensifier at all; if a speaker wishes to intensify the statement, they will use an adverb other than *hen* (like *zhen*, 'really').

[20] The abbreviations used in this section are from Li and Thompson (1981). NOM = nominalizer, CRS = currently relevant state, PVF = perfective aspect marker, CL = classifier. All other functional words are simply the pinyin of the Chinese word; all that are relevant to the discussion are explained in the text.

(42b) and (42c) seem to require a certain context. For a scalar (gradable) adjective like *pang*, (42b), which uses the nominalizing construction *shi…de* works best either when the state of fatness is being corrected or confirmed (I *am* fat) or in the context of a binary fat/thin alternative (*I could be fat or thin, but I'm fat*); for this reason, it has been viewed primarily as a focus marker (Choi, 2006). Note, though, that for a non-scalar adjective, like *shangdeng* 'top-quality', (42b) is the only option. (42c), on the other hand, appears primarily in parallelisms involving scalar adjectives:

(43) Wo pang danshi wo kuaile
I fat but I happy
I'm fat, but I'm happy.

Thus, it could be argued that in these cases some kind of ellipsis has occurred (the deletion of the semantically neutered *hen*), which additionally explains why absolute adjectives do not show either form (the intensifier *hen* selects for scalar forms, and is active even after ellipsis).

In (42d), *pang* is an adjectival verb, and the overall effect of the sentence is to communicate *I've gotten fat* (suggesting a recent change of events). If the communicative purpose is to express, for instance, frustration or dismay at recent developments, this seems like the form most likely to be used. On the other end of the spectrum, (42b) seems the most purely descriptive or objective (perhaps because it is the standard form for absolute adjectives, and thus there is no sense of subjective vagueness), with (42a) and (42c) falling somewhere in-between. All of this seems relevant to work by Hatzivassiloglou and Wiebe (2000), who investigated the detection of scalar adjectives and demonstrated their usefulness in subjectivity analysis. In the next section, we will look for both (objective) *shi…de* and (subjective) *le* in our corpus.

Reduplication of adjectives in Chinese has an intensifying effect, making the more description more vivid.

(44) ta xiande gao-gao-xing-xing-de
he seemed happ-happ-y-y-NOM
he seemed extremely happy

Note that reduplication of multisyllable adjectives involves reduplication of the individual syllables, contrasting with Chinese verb reduplication, which just involves repetition of the verb (and has the opposite semantics; repeated verbs are downplayed rather than intensified). This means that the appearance of these disyllabic words would be missed during automated analysis unless special measures are taken to identify them. That said, there are many scalar adjectives that cannot undergo this reduplication, and Li and Thompson suggest that in particular the number of disyllabic words that manifest this morphological change is fairly small (though many common words are included among them). One other interesting property of these reduplicated adjectives is that they cannot be additionally intensified, suggesting they might have a superlative reading, similar to Spanish *–ísimo*.

Reduplicated adjectives are especially common as adverbs; in fact, all monosyllabic adjectives, such as *man*, ‘slow,’ must be reduplicated or otherwise modified when used as adverbs (the intensification seems to be semantically bleached in this situation). There are two major kinds of open class adverbs in Chinese: manner adverbs, which tend to provide information about the manner of an action or the state of mind of the agent; and resultative adverbs, which provide information about the ultimate effect of an action. These two types of adverbs are distinguished by their syntax, with manner adverbs appearing before the verb, and resultative adverbs appearing after; in general adverbs are formed using adjectives linked to the verb with the particle *de* (the manner and resultative *de* are pronounced the same but written with different characters, and the manner *de* is usually optional).

(45) a. ta man-man-de zou hui jia
he slow-slow-DE walk back home
He walked home slowly.
b. ta xingfen-de pao hui jia
he excited-DE run back home
Excited, he ran home.
(46) a. zhouzi bei xi de-gan-gan-jing-jing
table BEI wash DE-clean-clean
The table was washed to a high shine.
b. ba wo kua DE-de-yi-wang-xing
BA me praise DE-pleased-with-self-forget-form
I was praised until I forgot myself (got full of myself)

Function words *bei* and *ba* are used to indicate passivity and causation, respectively. Note that many adjectives can appear in both positions, so it is not a lexical distinction; rather, it seems more to do with whether an action or relevant actors are being (objectively) described or being (subjectively) judged. For instance, manner adverbs are quite common in narrative texts, but a manner adverb could not be used to give an opinion about how well a book was written.

(47) a. Ni zhe-ben shu xie de-hen zhuanye
You this-CL book write DE-very expert
b.? Ni hen zhuanye-de xie-le zhe-ben shu
You very expert-DE write-PFV this-CL book
This book of yours was written expertly (with expertise).

(47b), the manner form, is extremely odd if your intention is to praise the writer of the book, and only makes sense if it forms part of a list of past actions (i.e., you expertly wrote this book, then tried to get it published to no avail), at which point the *hen zhuanye* is presented more as an integral (given) part of the writing, rather than as the opinion of the speaker. This phenomenon might also reflect the basic topic/focus ordering preferences (Gundel and Fretheim, 2004); if the goal is to provide an opinion, then that new information (the opinion) should appear later in the sentence. In any case, it seems that the sentiment relevant adverbials are more likely to appear after the verb.

(46b) above contains an example of a *chengyu*, a Chinese idiom. Most *chengyu* are four characters long, and, unlike English idioms (which are often regarded as little more than clichéd speech), Chinese idioms are central to the language, with thousands of idioms (in all major parts of speech) used in everyday communication, though they are particularly common in literary language. *Chengyu* are not subject to phrase-level syntax or morphology (unlike *kick the bucket* in English) and cannot generally be modified, however they often have an internal syntax and semantics (since they are essentially fossils of earlier forms of the language).

(48) a. ta yi-mao-bu-ba.
he one-hair-not-pull
he's very stingy.
b. wo shi ai-mo-neng-zhu-DE
I am love-touch-can-help-DE
I would like to help but I can't.
c. bie ji-yu-qiu-cheng
don't impatient-at-beg-complete
Don't be impatient for success.

(46b) and (48c) both demonstrate that Chinese idioms often contain transparently positive or negative characters which could be used to predict the polarity of the idioms, however the other examples are much trickier. For sentiment analysis, it might be a better strategy to include dictionaries with manually tagged idioms, assuming they appear often enough to warrant this attention.

The last set of features we will look at in this study are sentence-final (or mood) particles. As their name suggest, these closed-class items appear at the end of an utterance, and in spoken Chinese are used often to communicate information about speaker intention, listener knowledge, speaker-listener relation, etc. (Sun 2005). One clear (but for our purposes mostly uninteresting) example of a sentence-final particle is the question particle *ma*. The semantic effect of most of the other sentence final particles is not so straightforward. Table 37 lists common particles in modern Chinese as well as situations in which they are used. Note that this is a consolidation of multiple resources including Li and Thompson (1981), Sun (2005), Wei (1997), and the Unilang Wiki[21]; it contains more particles than any single source that I could find (for instance, Li and Thompson mention only 6!) including some that were added after being noted in the corpus; it excludes, though, particles that are clearly dialect-specific. For certain particles there is also a general lack of consensus with regards to their usage, here I have tried to be inclusive, see Lin (2005) for relevant discussion (including differences in usage based on gender).

[21] http://www.unilang.org/wiki/index.php/Sentence_final_particles

Particle(s)	**Usage**
吗(ma)	Used to change a declarative to an interrogative.
吧 (ba)	Signals a polite request, soliciting or offering approval, uncertainty, or alternatives.
呗(bei)	Indicates resignation, an ambivalent suggestion, or a situation that should be obvious to the listener (scornful)
呢(ne)	Used for alternative (*what about*..?) or rhetorical questions, to reinforce a declarative or emphasize continuity
呕/哦 (ou)	Used to urge, give a friendly warning, or provide emphasis
了(le)	Used to signal that a connection to previous discourse or a change of state, bounds the situation as relevant to the time of speaking.
嘛(ma)	Expresses a desire for a declarative statement to become true or be recognized by the listener as part of common ground.
啊(a)/呀(ya)/呐(na)/哇(wa)	Expresses enthusiasm, obviousness, doubt, and also used to soften orders, has various forms depending the preceding element.
咯(lo)	Used in indicate obviousness
喔(o)	Used to indicate surprise
哟(yo)	Used as an imperative or an exclamatory marker
啦(la)	Contracted form 了(le) +啊(a)
喽(lou)	Contracted form 了(le) +呕(ou)
罢了(bale)	Has the approximately same effect as *and that's all* in English, signals finality

Table 37: List of Sentence Final Particles

None of these particles could be said to be explicitly carrying an SO value (except perhaps *bei*, which is almost always negative to a certain extent), yet it is fairly easy to see how they could be relevant to sentiment analysis, as intensifiers, downplayers, irrealis blockers, stylistic features, or perhaps as indicators of the location of sentiment in the text (i.e, dimension 1 in the Biber classification scheme, see Chapter 4). However, these particles are primarily used in oral Chinese, a fact which is obvious from the form of their characters; nearly all of them have the 口(*kou)* radical ('mouth'), which indicates a connection to speech. It is evident that they would not be found in very formal texts, but it is an open question as to whether and to what extent they appear in online reviews written by the public. The corpus study in the next section will attempt to answer that question for all the features we have examined in this section.

5.3.2 Corpus Study

Following Li and Sun (2007), I collected a set of 761 reviews from echina.com, a Chinese travel-oriented website based in mainland China. I choose reviews from the general tourist destination section because I assumed that they would have the widest range of discussion—with people talking about the main tourist attraction(s), but also mentioning hotels, transportation, food, etc.—and I supposed they might have stronger opinions in general (since vacations often reflect a large investment of money for a relatively short period of time). The site allows each person posting to their forums to give a rating between 1 and 5 (at tenths of a point interval). I chose four "natural wonder" tourist attractions which had a lot of posts and somewhat mixed reviews (averages of between 2.6 and 3.2); I do not intend to focus on differences between positive and negative reviews, but I wanted to be sure that both types were well represented. The average length of a review was about 200 characters, however length ranged from a single Chinese character to over 5000. The few that were in English were manually removed.

I begin with particles, since they can be counted in the corpus automatically:

Particle(s)	**Count**
吗(ma)	35
吧 (ba)	111
呗(bei)	0
呢(ne)	48
了(le)	1917
嘛(ma)	24
啊(a)/呀(ya)/呐(na)/哇(wa)	131
哦(o)	27
咯(lo)	8
喔(o)	1
哟(yo)	6
啦(la)	12
喽(lou)	0
罢了(bale)	3

Table 38:Particle Counts in Chinese Vacation Corpus

Unfortunately, there is a slight complication to the numbers in Table 38 with respect to *le* particles: in addition to the sentence final particle, there is a *le* (same character) that serves as a verbal suffix indicating perfective aspect. Adding to this complexity is the fact that the *le* can be ambiguous when the verb is at the end of the sentence, and can even serve both purposes simultaneously. Also, the same character can be pronounced *liao* and serve as an integral part of the verb *understand*. In order to get an idea what percentage of *le* is the sentence-final type, I manually checked the first 20 unambiguous examples in the corpus; 7 were the sentence-final variety, which means that the average appearance of sentence-final *le* is probably around 1 per (200 character) review, still far more than any other sentence final particle.

Clearly, there are a number of sentence-final particles which appear regularly in the corpus; several (*le*, *ba*, *a, ne, and o*) appear enough that they might be worth special attention, or sentence-final particles in general might be considered as feature that is potentially indicative of relevant comment. For comparison, I looked at the descriptive summary provided on the website for each of the four tourist destinations; in 1500 characters, there was not a single sentence-final particle (the two appearances of *le* both unambiguously signaled perfective aspect). What is not clear, however, is whether the appearance or lack of these particles has more to do with register or genre: would we see these particles in any informal writing, or is the purpose of the text playing a key role as well?

For other features, it was necessary to manually extract examples from the texts. In 100 of the online posts (approx. 20,000 characters), I looked for lone adjectives introduced with *shi…de*, reduplication of adjectives and verbs, manner and resultative adverbs, and Chinese idioms. Of the four, the least commonly appearing feature was *shi* <adjective> *de*, I found only 4 examples which didn't involve intensification/downplaying either inside or outside the *shi…de*. Two are obviously descriptive (*it was self-serve* and *it was 4-star*) and one was descriptive (*it was not realistic*) in the context of a hypothetical situation where one tried to see everything on the mountain in one day. The forth example is rather interesting: the word *bucuo*. The word, which is clearly evaluative, literally means *no mistakes*, though it is clearly been lexicalized, since it can now be modified by *hen* (very). It can either mean *correct* (formally) or simply *pretty good* (Wei 1997). In the example that I found (and I found the same usage again latter in the corpus), it was used as a concession, i.e., *the scenery was good*, *but*…. Since the topic of discussion was a famous Chinese tourist attraction known for its spectacular scenery, I would argue that this is simply confirmation of common ground (as a precursor to criticism), and not really an attempt to advance an personal opinion. The use of *bucuo* is telling, because it has a downplaying/distancing quality, it is the sort of word used by teachers or managers who, in offering praise, simultaneously assert their status as an objective authority on a matter. Therefore, far from being a counterexample to *shi…de* as an objectivity marker, the *bucuo* example seems to add further weight to that argument.

In the 100 posts, I saw 16 examples of adjective reduplication and 7 examples of verb reduplication (there were also a few noun reduplications, which denote universal quantification). The verb reduplications mostly involved basic actions done casually (equivalent to *looking around, walking around, having a bite*), which might have a certain positive connotation independent of the actual verbs. The adjectives fell into two types: adjectives which were being used as (manner) adverbs, which, as suggested, rarely have as sentiment-relevant meaning component (e.g., *zaozao*, *early-early*, to do something well in advance) though they can effect intensification (*xiaoxiao*, *little-little*, a tiny bit); the more relevant kind of adjective is typified by *yuyucongcong* (*yucong* = lush, green) to describe a forest; the reduplication communicates the intensity of the experience in a way the base form does not. This is also a good example of a word which is not simply the sum of its parts; the most common usage of the character *yu* (郁) is in the word for depression.

There were very few resultative adverbs in the corpus. This could be due to the nature of the topic, which only rarely involved the opinionated discussion of human actions. There were two examples with *wan* (play, have fun) appearing with adjectives *shuang* (good-feeling) and *hao* (good), one example with *anpai* (arranged) and *hao* (i.e., not well arranged), and one example with *xiuxi* (rest) and *hao* (well rested) though in one case the context was hypothetical. More interesting are the manner adverbs (over 20), which generally appeared when the author chose to narrate the details of their trip, a fairly common occurrence. A number of the adjectives formed from these adverbs would be positive or negative in another context, but their usage was essentially never intended to express opinion. A good example is *zixi* (careful), which could be used to praise someone, but as a manner adverb rarely has that meaning (second example is from the corpus):

(49) a. ta zuo shi hen zixi
she do things very careful
She is a meticulous worker.
b. dajia yao zixi cha qingchu
Everyone must careful check clear
Everyone should fully check things out (before leaving).

Note that the *qingchu* (clear) (49) is a resultative verbal complement, not a resulative adverb; as it happens, resultative verbal complements also have the potential to disrupt sentiment analysis. Consider the three uses of *hao* (good) in (50):

(50) a.hao-hao gan
well do
Do (your job) well, work hard.
b. Gan-hao le
do-good CRS
(I'm) finished.
c. Gan de-hao
do well
Well done!

In (50a) the reduplicated *hao* forms a manner adverb, and communicates a hope or an intention, rather than reality. In (50b), *hao* functions as a verbal complement that indicates completion. In (50c), *hao* forms a resultative adverb; this is the only one of the three where opinion is being expressed.

Finally, we look at Chinese idioms. In the 100 posts searched, I found 57 idioms, or more than one every two posts, making them more frequent than any feature investigated except for *le*. The vast majority contributed sentiment, though depending on the discourse it was sometimes the case that effect of the sentiment was fairly localized. Most were unique, however one particular idiom *bu-xu-ci-xing* (not-empty-this-journey, or 'this trip was worth it') appeared 4 times, and another (‘deserving of its reputation’) appeared twice. Of those that contributed sentiment, I counted 15 idioms whose semantic orientation could probably be derived directly from the characters appearing in it, including a few that would only work if negation and/or constituency was properly handled. An example is *bu-wu-dao-li*, literally not (*bu*) without (*wu*) reasonableness (*daoli*), i.e., ‘not unreasonable,’ ‘justifiably,’ which would require identification of the consitutient *daoli* followed by not one but two layers of negation in order to get the correct (positive) interpretation. There were also several where the orientation derived from a compositional approach would be downright wrong, for instance *tianyanmiyu*, literally ‘sweet language honeyed speech,’ referring often to false flattery.

There is a publicly available dictionary of positive and negative terms in Chinese (Ku and Chen, 2007), and so I decided to test its coverage; of the 40-some sentiment-relevant (in my opinion) idioms that I encountered in 100 online reviews, only 5 appeared in their dictionaries. This is not to say that these dictionaries don’t contain a significant number of idioms (they clearly do), it is simply that the sheer number of idioms in Chinese (several thousand) makes it difficult to get anything approaching full coverage. It is apparent, however, that Chinese idioms are common, stable, fairly reliable indicators of sentiment: disregard them at your peril.

In this chapter, we have investigated expanding our semantic model into other languages, demonstrating the benefits of adaptation, but also some of the challenges. Many of the basic linguistic facts relevant to sentiment do not change from language to language, however there are various details in each language that demand special attention if the full potential of a more linguistically-motivated sentiment analysis model is to be realized.

Conclusion

Sentiment analysis is a multi-faceted problem, and the research presented here only begins to scratch the surface of a long-term research agenda. In hopes that this project can make a unique contribution in this area, we have explicitly adopted an approach which places us outside the machine learning mainstream, eschewing sheer computational horsepower in favor of more linguistically satisfying solutions. The results presented above could be viewed as encouraging or somewhat deflating, depending on one's expectations. It is seems clear, for instance, that we are unlikely to uncover a linguistic "silver bullet," a single addition to the model that will result in huge performance gains. Instead, improvement is likely to come incrementally, as more and more linguistic features are identified and integrated into the model.

Although there is a great deal of work left to do, much progress has already been made. In Chapter 1, we traced some of the various efforts to classify and quantify the language of emotion, a goal which has been the topic of research for more than half a century. In Chapter 2 we saw the major advancements that have been made in just the last few years since the explosion in computational research in the area of sentiment analysis. The SO Calculator, described and evaluated in Chapter 3, represents a synthesis of many of those ideas; we saw that its cross-domain performance hovers around 80% in the main binary classification task, and it also seems to capture well the scalar nature of opinion. Chapter 4 demonstrated the expandability of the SO calculator, supplementing it with an external machine learning modules for genre detection. In Chapter 5 it was adapted to one language, and we explored potential features for adaption to another.

Though I have emphasized the dichotomy between SO Calc semantic models and SVM machine classifiers, one major goal of this work is simply to demonstrate the model-independent complexity of the problem, the various layers that must be attended to regardless of the tools that are applied. It is possible, and indeed highly desirable, that a machine learning algorithm could discover many of the facts we have programmed into the SO Calculator, and perhaps some that we might not even notice. We should, however, hold any classifier to a standard that goes beyond percentage performance in a particular corpus; quite simply, what a classifier is doing should make linguistic sense. If it does not, then there is may be some sleight of hand involved, and we should proceed skeptically. In particular, if a model does not benefit at all from the addition of linguistic features that are clearly relevant to sentiment, then we should both examine those linguistic intuitions closely (checking the frequency of features and appearances in texts) as well as consider the possibility that the model is fundamentally flawed. Talking a modular approach, i.e., solving the problems one by one in functionally separate steps (rather than trying to build a one-size-fits-all classifier), seems the best way to ensure lasting progress.

As we decide where to focus our efforts, there is an obvious tension between high-level and low-level improvements, i.e., focusing on genre and discourse structure as compared to, say, modality or dictionary expansion. Faced with the relative sparseness of low-level features in the texts (a phenomenon we noted in Chapter 3), a better strategy might be to step back and examine larger units of text, to avoid getting bogged down in mere “semantics”; indeed, the more holistic approach of Chapter 4 resulted in a measurable gain. We returned, however, to those details in Chapter 5 with the discussion of other languages, in part because I do not believe it is possible to get a good start without them, nor is it possible to ignore them indefinitely. As we make improvements that allow us to better detect various parts of the text and identify key sentiment-bearing text spans, it becomes increasingly important that we can accurately interpret the sentiment that these key expressions contain. On the other hand, we might have a perfect semantic grasp of every individual expression in the text, and still fail to properly identify the overall polarity of the text in the case where there is too much semantic noise coming from textual units that are less relevant to the overall opinion. In short, a good model must get both the big picture and the details right.

There are clear theoretical and practical advantages to an approach that treats sentiment as a numerical value instantiated in individual words; such a model seems both psychological plausible and computationally tractable, presenting us with many options for integrating the effects of context—e.g., negation, intensification, modality, and repetition. But the successes of machine leaning models and the obvious importance of lexicon-independent factors in distinguishing between positive and negative text (see Section 2 of Chapter 4) suggest that we might need to further supplement the notion of SO, either with a (genre-specific) hybrid model or perhaps a different level of calculation that is not so word-dependent. How do we incorporate into our model the fact that punctuation and function words (stylistic features) are also powerful markers of sentiment? How can we integrate the notion of congruence or dissonance, where sentiment arises out of a particular combination of words? How can we make use of rhetorical patterns in text beyond the weighting of SO-valued words?

One way forward is the widening of our scope. Even though it can be identified as a unique semantic dimension, evaluation does not exist in a vacuum, and “sentiment” analysis must eventually expand to deal with the full range of human sentiment, including certainty and doubt, excitement and passivity, respectfulness and irreverence. Included in this would be a better sense of temporal placement of emotion, as we are presumably concerned with present opinion rather than future hopes or past preconceptions; in some cases, however, a pointed description of the past events or the suggestion of a course of action is as telling as an explicit statement of opinion. Then there is the problem of target: not only could identifying the topic of opinion help us in the same way as our description genre detector (we can ignore irrelevant topics), it can also as improve our interpretation of sentiment, in ways ranging from word sense disambiguation to the identification of metaphorical comment. We should also collect additional experimental evidence relevant to the model presented here, making sure that we stay grounded in psychological reality even as the complexity of the computational model increases.

References

Abbasi, Ahmed, Chen, Hsin-Hsi, and Salem, Arab. 2008. Sentiment Analysis in Multiple Language: Feature Selection for Opinion Classification in Web Forums. *ACM Transactions on Information Systems* 26.

Andreevskaia, Alina, and Bergler, Sabine. 2006. Semantic tag extraction from WordNet glosses. In *Proceedings of 5th International Conference on Language Resources and Evaluation (LREC)*, 413-416. Genoa, Italy.

Andreevskaia, Alina, and Bergler, Sabine. 2008. When Specialists and Generalists Work Together: Overcoming Domain Dependence in Sentiment Tagging. Paper presented at *ACL-08*, Columbus, Ohio.

Aue, Anthony, and Gamon, Michael. 2005. Customizing Sentiment Classifiers to New Domains: a Case Study. Paper presented at *International Conference on Recent Advances in Natural Language Processing*, Borovets, BG.

Bartlett, Jake, and Albright, Russ. 2008. Coming to a Theater Near You! Sentiment Classification Techniques Using SAS Text Miner. Paper presented at *SAS Global Forum 2008*.

Bautin, Mikhail, Vijayaren, Lohit, and Skiena, Steven. 2008. International Sentiment Analysis for News and Blogs.

Bestgen, Yves. 2008. Building Affective Lexicons from Specific Corpora for Automatic Sentiment Analysis. Paper presented at *LREC 2008*, Marrakech, Moracco.

Bieler, Heike, Dipper, Stefanie, and Stede, Manfred. 2007. Identifying formal and functional zones in film reviews. In *Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue*, 75-78. Antwerp, Belgium.

Bloom, Kenneth, Garg, Navendu, and Argamon, Shlomo. 2007. Extracting Appraisal Expressions. Paper presented at *NAACL HLT 2007*, Rochester, NY.

Boiy, Erik, Hens, Pieter, Deschacht, K, and Moens, Marie-Francine. 2007. Automatic sentiment analysis of on-line text. In *Proceedings of the 11th International Conference on Electronic Publishing*. Vienna, Austria.

Boucher, Jerry D., and Osgood, Charles E. 1969. The Pollyanna hypothesis. *Journal of Verbal Learning and Verbal Behaviour* 8:1-8.

Brill, Eric. 1992. A simple rule-based part-of-speech tagger. Paper presented at *ANLP-92*, Trento, IT.

Bruce, Rebecca F., and Wiebe, Janyce M. 2000. Recognizing subjectivity: A case study of manual tagging. *Natural Language Engineering* 5:187-205.

Chaovalit, Pinvadee, and Zhou, Lina. 2005. Movie review mining: A comparison between supervised and unsupervised classification approaches. In *Proceedings of the 38th Hawaii International Converence on System Science*.

Cho, Young Hwan, and Lee, Kong Jo. 2006. Automatic affect recognition using natural language processing techniques and manually built affect lexicon. *IEICE Transactions on Information and Systems* 89:2964-2971.

Choi, Kwok Tim. 2006. Chinese "SHI...DE" focus constructions: an Optimality-Theoretic proposal, Department of Linguistics, Simon Fraser University.

Church, Kenneth W., and Hanks, Patrick. 1989. Word association norms, mutal information and lexicography. Paper presented at *27th Annual Conference of the ACL*, New Brunswick, NJ.

Cohen, Jacob. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement* Vol.20:37-46.

Das, Sanjiv R., and Chen, Mike Y. 2001. Yahoo! for Amazon: Opinion extraction from small talk on the web. In *Proceedings of 8th Asia Pacific Finance Association Annual Conference*. Bangkok, Thailand.

Dave, Kushal, Lawrence, Steve, and Pennock, David M. 2003. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of the Twelfth International World Wide Web Conference (WWW 2003)*. Budapest, Hungary.

Di Eugenio, Barbara, and Glass, Michael. 2004. The Kappa statistic: a second look. *Computational Linguistics* 30:95-101.

Eggins, Suzanne. 2004. Genre: Context of culture in text. In *An Introduction to System Functional Linguistics*. London: Continuum.

Esuli, Andrea, and Sebastiani, Fabrizio. 2005. Determining the semantic orientation of terms through gloss classification. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*. Bremen, Germany.

Esuli, Andrea, and Sebastiani, Fabrizio. 2006. SentiWordNet: A publicly available lexical resource for opinion mining. In *Proceedings of 5th International Conference on Language Resources and Evaluation (LREC)*, 417-422. Genoa, Italy.

Finn, Aidan, and Kushmerick, Nicholas. 2003. Learning to classify documents according to genre. In *Proceedings of IJCAI Workshop on Computational Approaches to Text Style and Synthesis*. Acapulco, Mexico.

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

González, Trinidad, and Farrell, Joseph. 2001. *Composición práctica: Conversación y repaso*. New York: John Wiley & Sons.

Greenberg, Joseph H. 1996. Language Universals. In *Current Trends in Linguistics, III*, ed. T.A. Sebeok, 61-112. The Hangue: Mouton.

Grice, Paul. 1975. Logic and Conversation. In *Syntax and Semantics*, eds. Peter Cole and Jerry L. Morgan, 41-58. New York: Academic Press.

Gundel, Jeanette K., and Fretheim, Thorstein. 2004. Topic and focus. In *The Handbook of Pragmatics*, eds. Laurence Horn and G. Ward, 175-196. Malden, Mass: Blackwell.

Halliday, Michael A. 2004/1994. *An Introduction to Functional Grammar*. London: Edward Arnold.

Hatzivassiloglou, Vasileios, and McKeown, Kathleen. 1997. Predicting the semantic orientation of adjectives. In *Proceedings of 35th Meeting of the Association for Computational Linguistics*, 174-181. Madrid, Spain.

Hatzivassiloglou, Vasileios, and Wiebe, Janyce. 2000. Effects of adjective orientation and gradability on sentence subjectivity. Paper presented at *18th International Conference on Computational Linguistics*, New Brunswick, NJ.

Hearst, Marti. 1992. Direction-based text interpretation as an information access refinement. In *Text-Based Intelligent Systems: Current Research and Practice in Information Extraction and Retrieval*, ed. P. Jacobs. Mahwah, NJ: Lawrence Erlbaum Associates.

Hiroshi, Kanayama, Tetsuya, Kasukawa, and Hideo, Wantanabe. 2004. Deeper sentiment analysis using machine translation technology. Paper presented at *COLING '04*, Morristown, NJ, USA.

Holsti, Ole R. 1964. An Adaption of the 'General Inquirer' for the Systematic Analysis of Political Docments. *Behvaioral Science* 9:382-388.

Horn, Laurence. 1989. *A Natural History of Negation*. Chicago: University of Chicago Press.

Hu, Minqing, and Liu, Bing. 2004. Mining and summarizing customer reviews. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD-2004)*. Seattle, WA.

Hu, Yi, Duan, Jianyong, Chen, Xiaoming, Pei, Bingzhen, and Lu, Ruzhan. 2005. A New Method for Sentiment Classification in Text Retrieval. Paper presented at *IJCNLP 2005*.

Kaji, Nobuthiro, and Kitsuregawa, Masaru. 2007. Paper presented at *Joint Conference on Emperical Methods in Natural Language Processing and Computational Nautral Language Learning.*, Prague.

Kamps, Jaap, Marx, Maarten, Mokken, Robert J., and de Rijke, Maarten. 2004. Using WordNet to measure semantic orientation of adjectives. In *Proceedings of 4th International Conference on Language Resources and Evaluation (LREC 2004)*, 1115-1118. Lisbon, Portugal.

Kelly, Edward, and Stone, Philip J. 1975. *Computer Recognition of English Word Sense*: North-Holland Linguistic Series.

Kennedy, Alistair, and Inkpen, Diana. 2006. Sentiment classification of movie and product reviews using contextual valence shifters. *Computational Intelligence* 22:110-125.

Kilgarriff, Adam. 2007. Googleology is bad science. *Computational Linguistics* 33:147-151.

Kilma, Edward S. 1964. Negation in English. In *The Structure of Language*, ed. Jerry A. Fodor and Jerrold J. Katz, 246-323. Englewood Cliffs, N.J.: Prentice-Hall.

Knott, Alistair. 1996. A Data-Driven Methodology for Motivating a Set of Coherence Relations, Department of Artificial Intelligence, University of Edinburgh: Ph.D. dissertation.

Koppel, Moshe, and Schler, Jonathan. 2005. Using neutral examples for learning polarity. In *Proceedings of IJCAI 2005*. Edinburgh, Scotland.

Krippendorf, Klaus. 1980. *Content Analysis: An Introduction to Its Methodology*. Thousand Oaks, CA: Sage.

Ku, Lun-Wei, Wu, Tung-Ho, Lee, Li-Ying, and Chen, Hsin-Hsi. 2005. Construction of an Evaluation Corpus for Opinion Extraction. Paper presented at *NTCIR 2005*.

Ku, Lun-Wei, Liang, Yu-Ting, and Chen, Hsin-Hsi. 2006. Opinion extraction, summarization and tracking in news and blog corpora. In *Proceedings of the AAAI-2006 Spring Symposium on "Computational Approaches to Analyzing Weblogs"*, 100-107. Stanford, CA.

Ku, Lun-Wei, and Chen, Hsin-Hsi. 2007. Mining Opiions from the Web: Beyond Relavance Retrieval. *Journal of American Society for Information Science and Technology* 58:1838-1850.

Landauer, Thomas K., and Dumais, Susan T. 1997. A Solution to Plato's problem: The latent semantic analysis theory of the acquisition, induction, and representation of knowledge. *Psychological Review* 104:211-240.

Lerman, Kevin, Gilder, Ari, Dredze, Mark, and Perierra, Fernando. 2008. Reading the Markets: Forecasting Public Opinion of Political Candidates by News Analysis. Paper presented at *COLING '08*, Manchester.

Li, Charles N., and Thompson, Sandra A. 1981. *Manadarin Chinese: A Functional Reference Grammar*. Berkeley: University of California Press.

Li, Jun, and Sun, Maosong. 2007. Experimental study on sentiment classification of Chinese review using machine learning techniques. Paper presented at *IEEE-NLPKE2007*.

Lin, Huey Hanna. 2005. Contextualizing Linguistic Politeness in Chinese, The Ohio State University: Dissertation.

Lloyd, L., Mehler, A., and Shkiena, S. 2005. Lydia: A System for Large-Scale News Analysis. Paper presented at *12 Symp. of String Processing and Information Retrieval*.

Lu, Bin, Tsou, Benjamin K., and Kwong, Oi Yee. 2008. Supervised Approaches and Ensemble Techniques for Chinese Opinion Analysis at NTCIR-7. Paper presented at *NTCIR-7 Workshop Meeting*, Tokyo, Japan.

Mann, William C., and Thompson, Sandra A. 1988. Rhetorical Structure Theory: Toward a functional theory of text organization. *Text* 8:243-281.

Martin, James R. 1992. Macroproposals: meaning by degree. In *Discourse Description: Diverse linguistic analysis of a fund-rasing text*, eds. William C. Mann and Sandra A. Thompson. Amsterdam & Philadelphia: Benjamins.

Martin, James R. 1984. Language, register and genre. In *Children Writing:Reader*, ed. Frances Christie. Victoria: Deakin University Press.

Martin, James R., and White, Peter. 2005. *The Language of Evaluation*. New York: Palgrave.

Mihalcea, Rada, Banea, Carmen, and Weibe, Janice. 2007. Leaning Multilingual Subjective Language via Cross-Lingual Projections. Paper presented at *ACL '07*.

Miller, George E. 1990. WordNet: An on-line lexical databse. *International Journal of Lexicography* 3:235-312.

Mosier, Charles I. 1941. A psychomeric study of meaning. *Journal of Social Psychology* 13:123-140.

Mullen, Tony, and Collier, Nigel. 2004. Sentiment analysis using support vector machines with diverse information sources. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*. Barcelona, Spain.

Ng, Vincent, Dasgupta, Sajib, and Arifin, S. M. Niaz. 2006. Examining the Role of Linguistic Knowledge Sources in the Automatic Identification and Classification of Reviews. Paper presented at *COLING/ACL 2006*, Sydney.

Orasan, Constantin. 2003. PALinkA: a highly customizable tool for discourse annotation. Paper presented at *the 4th SIGdial Workshop on Discourse and Dialog*, Sapporo, Japan.

Osgood, Charles E., Suci, George J., and Tannenbaum, Percy H. 1957. *The Measurement of Meaning*. Urbana: University of Illinois Press.

Osgood, Charles E., and Richards, Meredith Martin. 1973. From Yang and Yin to *and* Or *but*. *Language* 49:380-412.

Pang, Bo, Lee, Lillian, and Vaithyanathan, Shivakumar. 2002. Thumbs up? Sentiment classification using Machine Learning techniques. In *Proceedings of Conference on Empirical Methods in NLP*, 79-86.

Pang, Bo, and Lee, Lillian. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of 42nd Meeting of the Association for Computational Linguistics*, 271-278. Barcelona, Spain.

Pang, Bo, and Lee, Lillian. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of ACL 2005*. Ann Arbor, MI.

Pang, Bo, and Lee, Lillian. 2008. Opinion Mining and Sentiment Analysis. *Foundations and Trends in Information Retrieval* 2:1-135.

Polanyi, Livia, and Zaenen, Annie. 2006. Contextual valence shifters. In *Computing Attitude and Affect in Text : Theory and Applications*, eds. James G. Shanahan, Yan Qu and Janyce Wiebe, 1-10. Dordrecht: Springer.

Potts, Christopher. 2007. The Expressive Dimension. *Theoretical Linguistics* 33:165-198.

Qiu, Guang, Liu, Kangmiao, Bu, Jiajun, Chen, Chun, and Kang, Zhiming. 2007. Extracting Opinion Topcis for Chinese Opinions using Dependence Grammar. Paper presented at *ADKDD ’07*, San Jose, California USA.

Quirk, Randolph, Greenbaum, Sidney, Leech, Geoffrey, and Svartvik, Jan. 1985a. *A comphrensive grammar of the English language*. London: Longman.

Quirk, Randolph, Greenbaum, Sidney, Leech, Geoffrey, and Svartvik, Jan. 1985b. *A Comprehensive Grammar of the English Language*. London: Longman.

Read, Jonathon, Hope, David, and Carroll, John. 2007. Annotating expressions of appraisal in English. Paper presented at *Proceedings of Linguistic Annotation Workshop, ACL 2007*, Prague, Czech Republic.

Rietveld, Toni, and van Hout, Roeland. 1993. *Statistical Techniques for the Study of Language and Language Behavior*. Berlin: Mouton de Gruyter.

Riloff, Ellen, Padwardhan, Siddharth, and Wiebe, Janyce. 2006. Feature Subsumption for Opinion Analysis. Paper presented at *2006 Conference on Empirical Methods in Natural Language Processing*, Sydney.

Russell, Bertrand. 1905. On Denoting. *Mind* 14:479-493.

Salvetti, Franco, Reichenbach, Christoph, and Lewis, Stephen. 2006. Opinion Polarity Indentification of Movie Reviews. In *Computing Affect and Attitude in Text: Theory and Applications*, eds. James G. Shanahan, Yan Qu and Janyce Wiebe, 303-316. Dordrecht: Springer.

Seki, Yohei, Evans, David Kirk, Ku, Lun-Wei, Chen, Hsin-Hsi, Kando, Noriko, and Lin, Chin-Yew. 2007. Overview of Opinion Analysis Pilot Task at NTCIR-6. Paper presented at *NTCIR-7 Workshop Meeting*, Tokyo, Japan.

Seki, Yohei, Evans, David Kirk, Ku, Lun-Wei, Sun, Le, Chen, Hsin-Hsi, and Kando, Noriko. 2008. Overview of the Multilingual Opinion Analysis Task at NTCIR-7. Paper presented at *NTCIR-7 Workshop Meeting*, Tokyo, Japan.

Shi, Jilin, and Zhu, Yinggui. 2006. *The Lexicon of Chinese Positive Words*: Sichuan Lexion Press.

Sim, Julius, and Wright, Chris C. 2005. The Kappa Statistic in Reliability Studies: Use, Interpretation, and Sample Size Requirements. *Physical Therapy* 85:257-268.

Spertus, Ellen. 1997. Smokey: Automatic recognition of hostile messages. In *Proceedings of AAAI/IAAI '97*, 1058-1065. Providence, RI.

Stegert, Gernot. 1993. *Filme rezensieren in Presse, Radio und Fernsehen*. Munich: TR-Verlagsunion.

Stone, Philip J., Dunphy, Dexter C., Smith, Marshall S., and Ogilvie, Daniel M. 1966. *The General Inquirer: A Computer Approach to Content Analysis*. Cambridge, MA: MIT Press.

Stone, Philip J. 1997. Thematic text analysis: New agendas for analyzing text content. In *Text Analysis for the Social Sciences*, ed. Carl Roberts. Mahwah, NJ: Lawrence Erlbaum.

Sun, Chaofen. 2006. *Chinese: A Linguistic Introduction*. New York: Cambridge University Press.

Taboada, Maite, and Grieve, Jack. 2004. Analyzing appraisal automatically. In *Proceedings of AAAI Spring Symposium on Exploring Attitude and Affect in Text (AAAI Technical Report SS-04-07)*, eds. Yan Qu, James G. Shanahan and Janyce Wiebe, 158-161. Stanford University, CA: AAAI Press.

Taboada, Maite, Anthony, Caroline, and Voll, Kimberly. 2006. Creating semantic orientation dictionaries. In *Proceedings of 5th International Conference on Language Resources and Evaluation (LREC)*, 427-432. Genoa, Italy.

Taboada, Maite, Gilles, Mary Anne, McFetridge, Paul, and Outtrim, Robert. 2008. Tracking literary reputation with text analysis tools. Paper presented at *Meeting of the Society for Digital Humanities*, Vancouver.

Takamura, Hiroya, Inui, Takashi, and Okumura, Manabu. 2005. Extracting semantic orientations of words using spin model. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL2005)*, 133-140. Ann Arbor.

Tan, Songbo, and Zhang, Jin. 2008. An empirical study of sentiment analysis for chinese documents. *Expert Systems with Applications* 34:2622-2629.

Titov, Ivan, and McDonald, Ryan. 2008. A Joint Model of Text and Aspect Ratings for Sentiment Summarization. Paper presented at *46th Meeting of the Association for Computational Linguistics*, Columbus, Ohio.

Tong, Richard M. 2001. An operational system for tracking opinions in on-line discussions. In *Working Notes of the ACM SIGIR 2001 Workshop on Operational Text Classification*, 1-6. New York, NY: ACM.

Turney, Peter. 2002. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of 40th Meeting of the Association for Computational Linguistics*, 417-424.

Turney, Peter, and Littman, Michael. 2002. Unsupervised learning of semantic orientation from a hundred-billion-word corpus: National Research Council of Canada.

Turney, Peter, and Littman, Michael. 2003. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems* 21:315-346.

Voll, Kimberly, and Taboada, Maite. 2007. Not all words are created equal: Extracting semantic orientation as a function of adjective relevance. In *Proceedings of the 20th Australian Joint Conference on Artificial Intelligence*, 337-346. Gold Coast, Australia.

Wan, Xiaojun. 2008. Using Bilingual Knowledge and Ensemble Techniques for Unsupervised Chinese Sentiment Analysis. Paper presented at *EM-NLP 2008*, Honolulu.

Wang, Guangwei, and Araki, Kenji. 2008. An unsupervised opinion mining approach for Japanese weblog reputation information using an improved SO-PMI algorithm. *IEICE Transactions on Information and Systems* 91:1033-1041.

Wang, Suge, Wei, Yingjie, Li, Deyu, Wu, Zhang, and Li, Wei. 2007. A Hybrid Method of Feature Selection for Chinese Text Sentiment Classification. Paper presented at *Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2007)*.

Wei, Dongya ed. 1997. *A Chinese-English Dictionary*. Beijing: Foreign Language Teaching and Research Press.

Whitelaw, Casey, Garg, Navendu, and Argamon, Shlomo. 2005. Using Appraisal groups for sentiment analysis. In *Proceedings of ACM SIGIR Conference on Information and Knowledge Management (CIKM 2005)*, 625-631. Bremen, Germany.

Wiebe, Janyce, Breck, Eric, Buckley, Chris, Cardie, Claire, Davis, Paul, Fraser, Bruce, Litman, Diane J., Pierce, David R., Riloff, Ellen, Wilson, Theresa, Day, David, and Maybury, Mark. 2003. Recognizing and organizing opinions expressed in the world press. In *Working Notes of the AAAI Spring Symposium in New Directions in Question Answering*, 12-19. Stanford, CA.

Wilson, Theresa, Wiebe, Janyce, and Hwa, Rebecca. 2004. Just how mad are you? Finding strong and weak opinion clauses. In *AAAI 2004*. San Jose, CA.

Wilson, Theresa, Wiebe, Janyce, and Hoffmann, Paul. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the 2005 Human Language Technology Conference and the Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP-05)*. Vancouver, Canada.

Witten, Ian H., and Frank, Eibe. 2005. *Data Mining: Practical Machine Learning Tools and Techniques*. San Francisco: Morgan Kaufmann.

Wu, Yun, Zhang, Yan, Luo, Si-ming, and Wang, Xiao-jie. 2007. Comprehensive Information Based Semantic Orientation Identification. Paper presented at *Natural Language Processing and Knowledge Engineering 2007*, Beijing.

Yang, Ling, and Zhu, Yinggui. 2006. *The Lexicon of Chinese Negative Words*.

Yao, Jianxin, Wu, Gengfeng, Liu, Jian, and Zheng, Yu. 2006. Using Bilingual Lexicon to Judge Sentiment Orientation of Chinese. Paper presented at *CIT '06*.

Ye, Qiang, Lin, Bin, and Li, Yi-Jun. 2005. Sentiment Classification for Chinese Reviews: A Comparison between SVM and Semantic Approaches. Paper presented at *Fourth International Conference on Machine Learning and Cybernetics*, Guangzhou.

Ye, Qiang, Shi, Wen, and Li, Yi-Jun. 2006. Sentiment Classification for Movie Reviews in Chinese by Improved Semantic Oriented Approach. Paper presented at *39th Hawaii International Conference on System Sciences*, Hawaii.

Yu, Hong, and Hatzivassiloglou, Vasileios. 2003. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 129-136. Sapporo, Japan.

Zagibalov, Taras. 2008. Basic Units for Chinese Opinionated Information Retrieval. *Journal of Siberian Federal University* Humanities & Social Sciences 1:115-123.

Appendices

Appendix 1: Full List of Intensifiers (English)

the_least	-3
less	-1.5
barely	-1.5
hardly	-1.5
almost	-1.5
not_too	-1.5
only	-0.5
a_little	-0.5
a_little_bit	-0.5
slightly	-0.5
marginally	-0.5
relatively	-0.3
mildly	-0.3
moderately	-0.3
somewhat	-0.3
partially	-0.3
a_bit	-0.3
to_some_extent	-0.25
to_a_certain_extent	-0.25
sort_of	-.3
sorta	-.3
kind_of	-.3
kinda	-.3
fairly	-0.2
pretty	-0.1
rather	-0.05
immediately	0.05
quite	0.1
perfectly	0.1
consistently	0.1
really	0.15
clearly	0.15
obviously	0.15
certainly	0.15
completely	0.15
definitely	0.15
absolutely	0.25
highly	0.25
very	0.25
truly	0.25
especially	0.25

particularly	0.25
significantly	0.25
noticeably	0.25
distinctively	0.25
frequently	0.25
awfully	0.25
totally	0.25
largely	0.25
fully	0.25
damn	0.25
intensively	0.25
downright	0.25
entirely	0.3
strongly	0.3
remarkably	0.3
majorly	0.3
amazingly	0.3
strikingly	0.3
stunningly	0.3
quintessentially	0.3
unusually	0.3
dramatically	0.3
intensely	0.3
extremely	0.35
so	0.35
incredibly	0.35
terribly	0.35
hugely	0.35
immensely	0.35
such	0.35
unbelievably	0.4
insanely	0.4
outrageously	0.4
radically	0.4
exceptionally	0.4
exceedingly	0.4
without_a_doubt	0.4
way	0.4
vastly	0.4
deeply	0.4
super	0.4
profoundly	0.4
universally	0.4
abundantly	0.4
infinitely	0.4
enormously	0.4
thoroughly	0.4
passionately	0.4

tremendously	0.4
ridiculously	0.4
obscenely	0.4
extraordinarily	0.5
spectacularly	0.5
phenomenally	0.5
monumentally	0.5
mind-bogglingly	0.5
utterly	0.5
more	-0.5
the_most	1
total	.5
monumental	.5
great	.5
huge	.5
tremendous	.5
complete	.5
absolute	.5
resounding	.5
drop_dead	.5
massive	.5
incredible	.5
such_a	.5
such_an	.5
utter	.3
clear	.3
clearer	.2
clearest	.5
big	.3
bigger	.2
biggest	.5
obvious	.3
serious	.3
deep	.3
deeper	.2
deepest	.5
considerable	.3
important	.3
extra	.3
major	.3
crucial	.3
high	.3
higher	.2
highest	.5
real	.2
true	.2
pure	.2
definite	.2

much	.3
small	-.3
smaller	-.2
smallest	-.5
minor	-.3
moderate	-.3
mild	-.3
slight	-.5
slightest	-.9
insignificant	-.5
inconsequential	-.5
low	-2
lower	-1.5
lowest	-3
few	-2
fewer	-1.5
fewest	-3
a_lot	.3
a_few	-.3
a_couple	-.3
a_couple_of	-.3
a_lot_of	.3
lots_of	.3
at_all	-.5
a_great_deal_of	.5
a_ton_of	.5
a_bunch_of	.5
a_certain_amount_of	-.2
some	-.2
a_little_bit_of	-.5
a_bit_of	-.5
a_bit_of_a	-.5
difficult_to	-1.5
hard_to	-1.5
tough_to	-1.5
nowhere_near	-3
not_all_that	-1.2
not_that	-1.5
out_of	-2

Appendix 2: Sample Text and SO Calculator Output

Plot Details: This opinion reveals major details about the movie's plot.
Mona Lisa Smile is a deck stacked in Julia Roberts' favor. The movie's premise is that every girl in a certain 1950's women's college is biding her time until she's lucky enough to find a man to provide for her. In like a California breeze sweeps Berkeley graduate Katherine Watson (Roberts), to blow the cobwebs off of these young girls' unused minds.

No doubt there were many repressed women in the pre-feminist era, but were so many of them gathered in one particular place? The girls at Wellesley College are Stepford Wives in training. They let men treat them brutishly (no physical abuse, mind you--the women just aren't allowed to think for themselves). Or they drink lots of booze and smoke up a storm. Or they sit around at night, practicing being spinsters (particularly Marcia Gay Harden in a really thankless role).

In fact, the only role more thankless than Harden's is that of Kirsten Dunst, so charming in Spider-Man and such a prig here. As Betty Warren--the school's unhappily married, McCarthy-like reporter--the sole point of Dunst's character is to make everyone as miserable as she is. Warren really takes passive-aggressiveness to an ethereal level.

But there are a lot of cracks in Katherine Watson's progressive thinking, too. First off, if she's such a smart thinker, why is Watson making time with a prof (Dominic West) who has a rep for sleeping with the students?

Secondly, there's the little speech that one of the students makes to Katherine near movie's end. In effect, the student says that since Katherine wants every woman to make a choice, she's made her choice to be a housewife, and what's wrong with that? From the moviemakers' point of view, the speech is meant to be ironic, but it actually leaks the ugly little secret that radical feminists don't want to hear: Another woman's choice doesn't always agree with your own.

Anyway, the movie gives its game away at about the halfway point, when the girls usher Katherine into their secret cult, just like in Dead Poets Society. How progressive can a feminist movie be when it's set in the '50s and yet steals its ideas from a lousy '80s men's movie?

Mona Lisa Smile is rated PG-13 for adult language and situations.

```
---------
Nouns:
-----
no physical abuse -2.0 + 3.0 (NEGATED)  = 1.0
a lot of cracks in -2.0 X 1.3 (INTENSIFIED) X 2.0 (HIGHLIGHTED)  X 1.5 (NEGATIVE) = -7.8
leaks -1.0 X 2.0 (HIGHLIGHTED)  X 1.5 (NEGATIVE) = -3.0
-----
Average SO: -3.26666666667
-----
Verbs:
-----
```

```
repressed -2.0 X 0 (QUESTION) = 0
steals -2.0 X 0 (QUESTION) X 0 (QUOTES) = 0
-----
Average SO: 0
-----
Adjectives:
-----
lucky 2.0  = 2.0
young 1.0  = 1.0
really thankless -2.0 X 1.15 (INTENSIFIED)  X 1.5 (NEGATIVE) = -3.45
more thankless -2.0 X 0.5 (INTENSIFIED) (COMPARATIVE) X 1/2 (REPEATED)  X 1.5 (NEGATIVE) = -0.75
so charming 4.0 X 1.35 (INTENSIFIED)  = 5.4
miserable -5.0  X 1.5 (NEGATIVE) = -7.5
ethereal 2.0  = 2.0
progressive 2.0 X 2.0 (HIGHLIGHTED)  = 4.0
such a smart 2.0 X 1.5 (INTENSIFIED) X 0 (QUESTION) = 0
little -1.0  X 1.5 (NEGATIVE) = -1.5
wrong -2.0 X 0 (QUESTION) = 0
ironic -2.0  X 1.5 (NEGATIVE) = -3.0
ugly -5.0 X 2.0 (HIGHLIGHTED)  X 1.5 (NEGATIVE) = -15.0
little -1.0 X 2.0 (HIGHLIGHTED) X 1/2 (REPEATED)  X 1.5 (NEGATIVE) = -1.5
radical -1.0 X 2.0 (HIGHLIGHTED)  X 1.5 (NEGATIVE) = -3.0
progressive 2.0 X 0 (QUESTION) X 1/2 (REPEATED) = 0
lousy -4.0 X 0 (QUESTION) X 0 (QUOTES) = 0
-----
Average SO: -1.25294117647
-----
Adverbs:
-----
unhappily -3.0  X 1.5 (NEGATIVE) = -4.5
enough 1.0  = 1.0
-----
Average SO: -1.75
-----
```

Total SO: -1.57272727273

Appendix 3: Full List of Discourse Features for Paragraph Genre Detection

;
:
!
?
,
(
[NN|NNS] (singular and plural nouns)
(_[NNP|NNPS] (proper nouns with left parentheses)
[NNP|NNPS]_) (proper nouns with right parentheses)
1stp (first person pronouns)
2ndp (second person pronouns)
3rdp (third person pronouns)
PRP$ (possessive pronouns)
POS (possessives)
appreciation
comparatives
conditionals
concluders
intensifiers
downplayers
negatives
cont:still (the word *still*, see Pang and Lee, 2002)
nesmod (necessity modal)
topicswitch
hedges
dempro (demonstrative pronouns)
causatives
it
JJ (adjectives)
judgement
time (adverbials)
place (adverbials)
posmod (possibility modals)
pred:will (predictive modals, these were split because they showed opposite tendencies)
pred:would
RB (adverbs)
Text_Position (0 – 1, paragraph i of n total at (i-1)/(n-1))
First_Paragraph (binary)
Last_Paragraph (binary)
VBN   (past participles)
VBD (past tense)
EX (existential *there*)
WDT (Wh-determiner)
WRB (Wh-adverb)

Appendix 4: Full List of Tags for Movie Zones Annotation Schema

Functional zones:
Describe-Plot
Describe-Character
Describe-Specific
Describe-General
Describe-Content
Describe+Comment-Plot
Describe+Comment-Actors+Characters
Describe+Comment-Specific
Describe+Comment-General
Describe+Comment-Content
Comment-Plot
Comment-Actors+Character
Comment-Specific
Comment-General
Comment-Overall
Quote
Background
Interpretation

Formal zones:
Tagline
Structure
Off-topic
Title
Title+Year
Runtime
Country+Year
Director
Genre
Audience-Restriction
Cast
Credits
Show-loc+Date
Misc-movie-info
Source
Author
Authorbio
Place
Date
Legal-notice
Misc-review-info
Rating