“I Can’t Recommend This Paper Highly Enough”: Valence-Shifted Sentences in Sentiment Classification

Logan Dillard

A thesis
submitted in partial fulfillment of the
requirements for the degree of

Master of Science

University of Washington

2007

Program Authorized to Offer Degree:
Institute of Technology - Tacoma

University of Washington
Graduate School

This is to certify that I have examined this copy of a master's thesis by

Logan Dillard

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Committee Members:

_____
Steve Hanks

_____
Isabelle Bichindaritz

Date:_____

University of Washington

Abstract

"I Can't Recommend This Paper Highly Enough":
Valence-Shifted Sentences in Sentiment Classification

Logan Dillard

Chair of the Supervisory Committee:
Professor Steve Hanks
Computing and Software Systems, Institute of Technology

Sentiment Classification, or discovering the sentiment of opinions expressed in unstructured text, has been the subject of much recent work. Approaches to sentiment classification use the standard classification algorithms and set of n-grams tokenization typical of other text classification tasks.

While current sentiment classification systems perform well on the majority of text, there remain particular linguistic features that present significant problems to today's state-of-the-art systems. One such linguistic feature is a Valence Shifter (VS), a word that alters the sentiment expressed by other words. The primary category of valence shifter in sentiment classification is negation.

This study analyzes the frequency of valence shifters and their effect on the performance of current sentiment classification systems in the domain of product reviews. It also categorizes valence shifters by syntactic categories, finding the following four categories to encompass the entire set: negated verb phrases, negated noun phrases, negated adjective phrases, and modal phrases. Valence shifters are found to be relatively common in the product reviews that compose this study's corpus, appearing in 15% of all sentences. They are also found to be particularly problematic for the classifier, especially in terms of Improper Class (IC) precision errors (in which a sentence with positive sentiment is classified as negative or vice-versa). A simple technique for handling valence shifters is found to perform well, despite the fact that it does not address and exploit the complicated syntax commonly found in valence-shifted sentences. This technique leads to a reduction in IC errors of 61% in VS sentences, and a 29% reduction in IC errors over the entire corpus.

TABLE OF CONTENTS

	Page
List of Tables	ii
1 Introduction	1
2 Sentiment Classification	3
3 Solution Technologies	6
4 Valence Shifters in Sentiment Classification	9
5 Related Work	12
6 The Impact of Valence Shifters in Sentiment Classification	15
7 Strategies for handling negation	18
8 Results of Negation-Handling Approaches	23
9 Discussion of Results	28
10 Conclusion and Future Work	31
References	33

LIST OF TABLES

Table Number	Page
1. Frequency of Valence Shifters by Valence-Shifter Type	16
2. Frequency of Valence Shifters by Sentiment Category	16
3. The Baseline Classifier's Error Rates on Different Categories of VS	17
4. Results of All Approaches on the 10-Fold Cross-Validated Set	24
5. Results of All Approaches on the Set of All VS Sentences	25
6. Results of All Approaches on the Set of Non-VS Sentences	25
7. Results on Negated Verb Phrases	26
8. Results on Negated Noun Phrases	26
9. Results on Negated Adjective Phrases	26
10. Results on Modal Phrases	27
11. Results of the Speed Test for All Approaches	27
12. Class-Conditional Probabilities for Words and their Negated Counterparts	30

ACKNOWLEDGEMENTS

The author wishes to express his sincere appreciation to the following parties who have been instrumental in work to produce this thesis: Steve Hanks for his mentorship, Michael Gamon for his advice and support, Microsoft Corporation for generous funding, and Cristoph Ponath for coordination and support from Microsoft.

1 Introduction

Sentiment Classification aims to determine the orientation of opinions expressed in unstructured text. For example, a sentiment classification system would classify the sentence "*This camera is great*" as expressing positive sentiment, the sentence "*This camera is a piece of junk*" as expressing negative sentiment, and the sentence "*I bought this camera two weeks ago*" as not expressing sentiment.

Sentiment classification is useful for a variety of reasons. It allows automatic summarization of the multitude of opinions on the web about products, brands, businesses, restaurants, events, cultural and artistic works, political candidates, etc. Thus it has informational value to people who might be interested in these items, as well as commercial value to vendors and producers of these items.

This study addresses a specific issue in sentiment classification, that of valence shifters (VS). Valence refers to the positive or negative emotional charge (sentiment) of a word, and thus a valence shifter is a word that alters the emotional charge of another word. Negation is a common example of a valence shifter, and one that this study focuses on. Valence shifters present a significant problem to sentiment classification, since they cause words that typically have a given sentiment to actually carry the opposite sentiment in the context of the valence shifter. For example, though the phrase "*very good*" typically indicates positive sentiment, when negated it may indicate negative sentiment, as in the phrase "*not very good*."

This study makes the following contributions: an analysis of valence shifters in sentiment classification in terms of frequency and contribution to errors, and an evaluation of approaches to solving the VS problem, offering a simple but surprisingly effective solution.

The outline of the rest of this paper is as follows: Section 2 describes sentiment classification in more detail; Section 3 presents current solution technologies; Section 4 discusses valence shifters; Section 5 presents related work; Section 6 examines the impact of VS in sentiment classification; Section 7 describes strategies for handling VS; Section 8 presents results of VS-handing strategies;

Section 9 discusses these results; finally Section 10 concludes and discusses future work.

2 Sentiment Classification

Though sentiment classification is related to the task of text categorization in Information Retrieval, it also has important differences. Typical text categorization tasks categorize documents according to topic. While words and phrases that represent different topics are likely to be semantically independent, the sentiment of a document is determined by the interaction of all sentiment-bearing words or phrases within the document. For example, a document that mentions "*three-pointers*" and "*slam dunks*" in one paragraph and then goes on to mention "*touchdowns*" and "*field goals*" in several other paragraphs is likely to simply discuss both basketball and football. In contrast, a document that begins with a paragraph containing phrases like "*above average*" and "*ample*," but then goes on to include many paragraphs with phrases like "*disappointing*" and "*poor quality*" may have negative sentiment overall.

Furthermore, a document's sentiment is often expressed through the composition of multiple words, which may include various parts of speech and span multiple phrases, while topics are usually referenced by individual words or contiguous multi-word terms. For example, the sentiment of the following phrases is created through the composition of non-consecutive words:

> "*I am not able to highly recommend this product*,"
> "*This is not the camera for you*."

In the following sentence, sentiment is expressed through multiple phrases that can only be properly interpreted in the context of what precedes them:

> "*It seems to crash every five minutes, so if that's what you're looking for, then you should definitely buy this*."

Sentiment classification has been performed on different levels of granularity, from documents that include many paragraphs to individual sentences. Sentiment information gathered at the sentence level can expose which aspects of the subject the

author feels positively or negatively about, rather than simply reporting sentiment for the subject itself. Depending on the domain, the sentence level may also be a more appropriate level for measuring sentiment, as many documents contain both positive and negative sentiment.

Many studies on sentiment classification use accuracy as their chief performance metric, while others use precision, recall, and f-measure. Accuracy assumes equal costs for all types of errors, which is likely to be inappropriate in sentiment classification, especially if the system is to present its results on individual documents to users. Precision and recall are better on this front, as they separate false positives from false negatives. F-measure, which is a harmonic mean of precision and recall, again blurs these lines.

However, sentiment classification is not a problem of Boolean classification, but rather a multi-class classification problem. Many studies (including this one) classify sentiment into the categories of positive, negative, mixed, and none. Since errors between different classes are likely to have different costs, a more specific system of metrics may be more appropriate.

The authors of this study start with the observation that there are three types of errors in sentiment classification: improper class, jump to judgment, and lost opportunity. An “improper class” error (IC) is one in which an opinion of one sentiment class (positive or negative) is classified as the opposite opinion class (negative or positive, respectively). A “jump to judgment” error (JJ) is one in which a sentiment-free opinion is classified as having either positive or negative sentiment. Finally, a “lost opportunity” error (LO) is the classification of a positive or negative opinion as belonging to the no-sentiment class. The distinction between these error types is important for an application that presents opinions to users with the machine-determined label. LO errors affect the amount of data that is generated, which may be of high or low concern depending on the application and the performance of the system. JJ errors are often not distressing to users, as they tend to be made on opinions that different humans may label in different ways. Finally, IC is the most

serious category of error, as these errors usually stand out to a user as egregiously incorrect.

3 Solution Technologies

Techniques successful in other text categorization tasks are not necessarily successful in sentiment classification. A typical topic-based text categorization task might use a simple keyword-based approach to classify text based on topic (e.g. high incidence of words like "*touchdown*" and "*half-time*" means a document is about sports, while the presence of phrases like "*stock price*" and "*merger*" indicates that a document is about business). Such keywords can be chosen by relative document frequency. This approach works well because the mere presence of these terms indicates the topic of the text.

However, simple keyword-based approaches do not tend to perform well on sentiment classification [Rimon]. To use an example from [Lee], a keyword-based approach might use the phrase "*great deal*" as an indicator of positive sentiment. However, due to the richness and complexity of unstructured natural language, term presence does not necessarily indicate sentiment, as the following sentences show [Lee]:

- This laptop is a great deal.
- A great deal of media attention surrounded the release of the new laptop model.
- If you think this laptop is a great deal, I've got a nice bridge you might be interested in.

These sentences have positive sentiment, no sentiment, and negative sentiment, respectively.

Current state-of-the-art sentiment classification systems are typically based on a machine-learning component, often a Naïve Bayes model or SVM (as used in [Pang et al.], [Dave et al.], and [Kennedy and Inkpen]). The features for these models are produced from the text using several preprocessing techniques, including stop-word removal, punctuation filtering, stemming, and n-gram tokenization. Preprocessing is performed in order to create meaningful features for the classifier that both distinguish between words and phrases with different meanings and conflate those

with similar meanings. The rest of this section explains these techniques in more detail.

A stop-word list is a list of words known to be irrelevant to the classification task at hand. Stop-word lists for all kinds of text categorization are typically composted of function words, such as “I,” “it,”, and “the.” Stop-word lists for sentiment classification may include other common words that are not associated with any sentiment (or lack thereof). A text classification system using a stop-word list would remove all words contained in the stop-word list from the document during preprocessing. Punctuation may be removed for similar reasons.

Another common preprocessing step is Stemming. Stemming transforms words to a base form, often removing suffixes that denote inflection, conjugation, plurality, etc. This has the benefits of reducing the amount of data required for training, thus increasing coverage, and also to reduce the feature space, which reduces processing and memory requirements. The training data requirements are lessened by generalizing the information learned from all morphological forms of a word. For example, a stemmer would transform the words “*failed*,” “*failing*,” and “*fails*” all to the word “*fail*.” This would allow the classifier to learn that “*failing*” is negative a term by only seeing other variants of the word.

A common way to produce features in text processing is to use N-gram Tokenization. N-gram tokenization creates tokens out of each set of N consecutive words. Common sizes of N are 1 (unigrams), 2 (bigrams), and 3 (trigrams). N-gram tokenization using unigrams, bigrams, and trigrams for the sentence “This camera is great” would yield the following tokens (a stop-word list is not used in this example for clarity of explanation):

- unigrams: this, camera, is, great
- bigrams: this_camera, camera_is, is_great
- trigrams: this_camera_is, camera_is_great

Additional placeholders may also be inserted at the beginning and end of the sentence so that n-grams of size greater than 1 capture the individual words at the beginning and end of the sentence. For example, bigram tokens from the above sentence with added placeholder tokens would include the tokens null_this, and great_null.

Feature selection is often performed to reduce the feature space. Features often are retained based on frequency and strength of statistical association with a class. For example, a sentiment classification system may discard all tokens that appear fewer than three times and have a statistical significance of having a non-random distribution of $p < 0.05$. Log-likelihood ratio (LLR) is a common way of measuring how a feature's distribution differs from a random distribution. LLR compares a feature's distribution among classes to the frequency of the classes, producing a Chi-squared measure of statistical significance [Dunning].

Naïve Bayes (NB) classifiers and SVMs (Support Vector Machines) are general-purpose machine-learning classifiers. NB uses Bayes' rule to compute posterior probabilities for each class, and assumes conditional independence among features. SVMs are based on relatively recent statistical learning research [Vapnik]. They map input features to an augmented feature space in which they draw a linear decision boundary. This linear boundary in augmented feature space maps to a nonlinear boundary in input feature space, resulting in a classifier with an arbitrarily-shaped decision boundary. Though SVMs are generally considered the best performers on many classification tasks (including text processing tasks), the simpler NB often provides competitive performance.

This study uses a baseline sentiment classification system based on an NB model. The system filters stop-list words, removes punctuation, uses the Porter stemming algorithm [Porter], and tokenizes into unigrams, bigrams, and trigrams. It then eliminates tokens that occur fewer than two times in the corpus or which have a statistical significance (based on LLR) of less than a fixed threshold.

4 Valence Shifters in Sentiment Classification

The purpose of this study is to examine valence shifters in sentiment classification at the sentence level with the aim of reducing the rate of serious errors: those of "improper class" (IC). This study focuses on those valence shifters that invert or void the sentiment of other words, not those that merely augment or diminish sentiment without changing its polarity. Examples include negation, which tends to be marked by words like "not," "no," and "never," and modals, which include conditionals with "if," and other phrases marked by words like "would," "should," and "could." Valence shifters are discussed at length in [Polanyi and Zaenen].

This study shows that valence shifters (VS), of which negation is the primary category, are a significant problem in sentiment classification. A simple example of this problem is the following sentence:

> "*This is not the best camera*."

The typical approach to sentiment classification may react in a strongly positive way to the terms "*best*" and "*best camera*," though in the sentence these terms are negated by the word "*not*," and thus they indicate negative sentiment.

Not all instances of negated sentiment are as straightforward as the example above. Another example of negation shows the difficulty of the problem:

> "*The overly detailed approach makes it a hard book to recommend enthusiastically*" [Rimon]

Again, the sentence contains a sub-phrase ("*recommend enthusiastically*") that is likely to produce a strongly positive classification, and in this case the negation is more subtle. Surprisingly few sentences in real reviews are written in the straightforward way of the first example, especially when they express negative sentiment [Rimon, Lee]. Negation is more significant when performing sentiment classification on the sentence-level than on the document-level, as the sentiment

computed for the document as a whole can be correct even if a number of sentences are classified incorrectly due to negation.

Modals tend to be even more difficult to classify. Sentences like

"*I would like to highly recommend this product, but I'm afraid that I can't*,"

do not provide a consistent pattern, which can be seen by contrasting the previous sentence with the following:

"*I would like to highly recommend this product to anyone who doesn't already have one*."

Furthermore, conditionals frequently require world knowledge to interpret. To interpret the sentiment of a sentence like

"*Well, it's a good deal if you want a $194.99 coaster*,"

one must understand that $194.99 is a bad price for a coaster, and also that someone reading a consumer electronics review is probably not interested in using the product as a coaster.

Natural language processing (NLP) provides a set of techniques that seem potentially useful in solving the problem of VS in sentiment classification. If each sentence can be fully parsed using traditional NLP techniques, then one would imagine that negations could be recognized and their targets determined. However, standard NLP systems have also had difficulties with negation. The NLP-based sentiment classification system developed by Nasukawa (which will be discussed further in Section 5) fails on sentences similar to the following two:

"*It's not that it's a bad camera*," and
"*It's difficult to take a bad picture with this camera*" [Nasukawa].

NLP systems also have trouble interpreting improper grammar side-by-side with proper grammar, as is often found in informal writing like product reviews. Furthermore, NLP systems are often domain-specific, which could be a problem for a large-scale sentiment classification system.

5 Related Work

The techniques involved in sentiment classification come from three subfields of artificial intelligence: information retrieval (IR), machine learning (ML), and natural language processing (NLP). IR searches for documents or information in documents that is relevant to a given user, which often means being classified as a member of a category or relevant to a specific query. Web-based search engines are probably the most well-known applications of IR. ML is a broad subfield concerned with the development of algorithms and techniques that allow computer systems to improve their performance based on experience or information, often using statistical methods. NLP is concerned with developing computer systems that understand natural human language.

Sentiment classification is a relatively new area. One of the earliest papers on the topic was [Pang et al.]. Many studies have performed sentiment classification on an entire document (see [Dave et al.], [Turney], and [Pang et al.] among others), though some focus on individual sentences (see [Yi], [Nasukawa], and [Gamon et al.]), which is the path this project takes.

Many approaches to sentiment classification focus on statistical and machine learning techniques (such as [Pang et al.] and [Dave et al.]), while some favor NLP techniques (such as [Nasukawa] and [Yi]). The system developed by Nasukawa analyzes sentiment by performing syntactic parsing and referencing a sentiment lexicon [Nasukawa]. While this system achieves high precision, its recall is very low even within its training domain. [Yi] uses these systems as well as a sentiment pattern database. As mentioned previously, the systems in [Pang et al.] and [Dave et al.] use n-gram models with Naïve Bayes and SVM classifiers.

Negation has been mentioned as a “potentially important” problem [Pang et al.], both in studies that take an ML approach and in those that take an NLP approach (see [Pang et al.], [Dave et al.], and [Das and Chen] for an ML approach, and [Nasukawa] for an NLP approach). It was addressed specifically by Kennedy and Inkpen, who confirmed its significance [Kennedy and Inkpen]. They were able to achieve small but statistically significant improvements in classification by taking

valence shifters into account. While the study by Kennedy and Inkpen included not only negation, but also intensifiers (e.g. “very”) and diminishers (e.g. “hardly”) to help capture subtle valence shift, they reported that negation terms contribute a lot to the improvement, while intensifiers and diminishers contribute less.

Previous approaches to negation in sentiment classification involve tagging words following a negation word as being negated [Pang et al., Dave et al., Kennedy and Inkpen]. For example, the sentence

“*This is not the best camera*,”

would be changed to

“*This is NOTthe NOTbest NOTcamera*.”

Pang et al. tag words for negation until the next punctuation, reporting that it has a negligible, but on average positive, effect [Pang]. Dave et al. “mark all words following [a negation word] as negated”, reporting that the implementation actually hurts performance [Dave]. Kennedy and Inkpen included shallow parsing to determine the scope of the negation, and tagged those words inside the scope as negated [Kennedy and Inkpen]. While Pang and Kennedy work with movie reviews, Dave et al. use product reviews, like this study. None report significant differences in performance between domains.

A different approach to negation has been used in the domain of medical concept detection. Medical concept detection is the task of determining whether free-text medical documents, such as discharge summaries or surgery notes, contain references to certain medical concepts. This may be done in order to index documents for retrieval based on relevance to specified concepts. Mutalik et al. use context-free grammars to reliably identify negation of medical concepts, resulting in improvements for medical concept detection [Mutalik et al.]. Concept detection is more similar to the task of text categorization than to sentiment classification, and

thus the technique found to be effective in [Mutalik et al.] is unlikely to perform well in the context of this study. This topic is discussed further in the Discussion (Section 9).

6 The Impact of Valence Shifters in Sentiment Classification

This study investigates the hypothesis that valence shifters (VS) pose a significant problem in sentiment classification. This section includes a categorization of valence shifters found in the computer and electronics product review domain and the frequency of each type. It also includes an evaluation of the performance of the baseline classifier, as described in Section 3, on sentences from each of these VS types and from non-valence-shifted sentences from the corpus.

This study uses a corpus of sentences from product reviews in the "computers and electronics" domain. The corpus contains 22,270 sentences with cross-validated human-labeled sentiment, each of which was labeled as having either positive, negative, neutral, or mixed sentiment. 2701 of these 22,270 sentences were labeled for valence shifters by the authors of this study.

We divided the VS sentences into two categories, those containing negation and those containing modals. Negation can be further divided into the following three subcategories:

- negated verb phrases (NVP),
- negated noun phrases (NNP),
- and negated adjective phrases (NAP).

This study does not subcategorize modals. Most modals found in the corpus are conditionals often marked by the word "*if*." Others include phrases in modalities of what is not the case but what could, would, or should be. Common phrases include "*would have been nice*," "*would love to have*," and "*could use* [a desirable feature]."

Examples of each VS category are as follows:

- NVP – "*Therefore, I am not able to highly recommend this camera*"
- NNP – "*No major problems*"
- NAP – "*Not very reliable*"
- Modal – "*Great camera, if you like your pictures off-center*"

Table 1 shows the frequency of valence shifters found in the corpus by humans, separated by VS type. Negated verb phrases (NVP) are the most common, followed by negated noun phrases (NNP), then negated adjective phrases (NAP), and finally modals.

Table 1. Frequency of valence shifters by valence-shifter type. Percentages of VS types do not sum to the figure reported for all VS because some sentences contain multiple types of VS.

VS Type	Corpus	VS sentences
Negated VP	7.4%	49.4%
Negated NP	3.9%	25.9%
Negated ADJP	2.9%	19.3%
Modals	1.2%	8.1%
Total	15.0%	100.0%

Table 2 shows the frequency of valence shifters in each sentiment category. VS sentences are far more frequent in negative- and mixed-sentiment sentences than in positive-sentiment sentences. This may partly explain the poorer performance on the negative and mixed categories that sentiment classification systems display. VS is rare in sentences with no sentiment.

Table 2. Frequency of valence shifters by sentiment category.

Sentiment Category	VS frequency
All Sentiment	15.0%
Positive	9.6%
Negative	38.8%
Mixed	33.3%
None	1.7%

Table 3 shows the baseline classifier's error rates on different categories of VS. IC and LO errors are shown, but JJ errors are omitted. To review from Section 2, an IC error is one in which an opinion of one sentiment class (positive or negative) is classified as the opposite opinion class (negative or positive, respectively). A JJ error

is one in which a sentiment-free opinion is classified as having either positive or negative sentiment. Finally, an LO error is the classification of a positive or negative opinion as belonging to the no-sentiment class. Though JJ errors are more common among VS sentences, the difference is not as marked as that of other errors. IC errors are over four times more common in VS sentences than non-VS sentences, and LO errors are 40% more common in VS sentences than non-VS sentences. By combining the frequency information with the error information, it can be seen that VS sentences account for 44% of all IC errors, though they comprise just 15% of the corpus. Furthermore, it is possible that the IC error rate on non-VS sentences would be even lower than that reported in this study if the training set did not also include VS sentences.

Table 3. The baseline classifier's error rates on different categories of VS. IC and LO errors are shown.

Category	IC Rate	LO Rate
Labeled Corpus	2.46%	38.01%
No VS	1.71%	34.15%
All VS	7.50%	53.26%
Negated VP	6.42%	48.09%
Negated NP	3.77%	51.55%
Negated ADJP	16.67%	62.30%
Modals	13.33%	68.00%

The preceding data confirm that valence shifters are an important problem in sentiment classification. They are present in a significant fraction of all sentences (15%). They have error rates far higher than the non-VS corpus and account for nearly half of all the serious (IC) errors.

7 Strategies for handling negation

It was shown in the previous section that valence shifters cause a significant number of errors in sentiment classification, and their effect is most pronounced in terms of "improper class" (IC) errors. If the effect of valence shifters were removed or reduced, this could significantly improve the performance of sentiment classification systems.

The N-gram tokenization approach used by the baseline system is unable to account for the effects of valence shifters. N-gram tokenization removes syntactic information that is necessary for properly interpreting the sentiment of valence-shifted sentences. While the inclusion of bigrams and trigrams does add some structural information, this may be undermined by the presence of unigrams. This notwithstanding, excluding unigrams reduces system performance. Furthermore, syntactic information is completely lost beyond a distance of N=3 words.

Two examples of negated sentences and their semantic interpretation serve to illustrate how the N-gram approach looses necessary structural information (in the examples, bigrams are used). In the sentence

"*Not a very good product*,"

the valence shifter "*not*" operates on the noun phrase "*a very good product*." The semantic interpretation is

"*a (not very good) product*,"

which is essentially equivalent to the sentence

"*a bad product*."

In contrast, the bigram tokenization approach produces the following tokens:

not_a, a_very, very_good, and good_product.

The only tokens likely to have strong sentiment are very_good and good_product, which both have strongly positive sentiment.

In the double-negated sentence

"*Didn't find anything I didn't like*,"

the second negator operates on "*like*," and the first operates on the verb phrase that makes up the rest of the sentence. The semantic interpretation is

"*there is nothing that I found and did not like*,"

which is equivalent to the sentence

"*everything I found I liked*."

This can be accurately approximated by the sentence

"*I liked everything*."

In contrast, the bigrams produced from this sentence are the following:

did_not, not_find, find_anything, anything_I, I_did, did_not, and not_like.

The only token likely to have strong sentiment is not_like, which has strongly negative sentiment.

The approach taken by this study in order to improve performance on valence-shifted sentences has been to attempt to salvage the n-gram tokenization approach by augmenting the tokenization process to account for valence shifters. Thus, this

study's VS component is implemented as a preprocessing step that processes valence-shifted phrases into a form that is stable with respect to n-gram tokenization.

This study has developed two techniques that adopt this approach. The first is based on specific syntactic patterns, including part-of-speech information and specific words. The second technique is to apply a negation tag to all words closely following a negation word, similar to the approach used by [Pang et al.] and [Dave et al.]. It was expected that the first technique would yield high precision but low coverage, since it is composed of narrow-purpose, manually discovered rules. In contrast, the second technique was expected to yield high coverage but lower precision, since it is an automatic approach that reacts to all incidences of negation.

Developing the pattern-based approach was carried out by manually analyzing sentences to discover patterns and regularities. It was hypothesized that a relatively small number of patterns may cover the majority of problematic VS sentences. Patterns were developed independently for each category of VS. The following represents a typical pattern:

"not [article] [adjective] problem" → "noproblem."

A complete list of the patterns developed in this approach can be found at the end of this section.

Several options were tried for the technique of applying a negation tag to all words within the scope of the negation. The chief area of experimentation was the method of determining the scope of the negation word. The techniques tried for determining negation scope are as follows:

- NLP-based shallow parsing ("chunking") to determine the end of the phrase,
- manually created part-of-speech-based heuristics to determine the end of the phrase,
- a fixed window for all negation words (e.g., 3 words following the negation word),

- and fixed windows of different sizes for each negation word ("custom windows").

Ending at punctuation was also tried (as in [Pang et al.]) in combination with the fixed-window technique. Another experiment compared the effects of using a single tag for all negations versus distinct tags for each negation word (e.g. "*not*," "*no*," and "*never*"),

Additionally, a small number of preprocessing rules were manually discovered to aid the automatic negation tagging technique. These preprocessing rules fall into two categories, those that identify an instance of negation that doesn't use a typical "negation word" like "*not*," "*no*," or "*never*," and those that identify a false negation or a negation with reduced scope. An example of the first category is "*failed to*" in the sentence

"*the camera failed to work as advertised*,"

which is equivalent to the sentence

"*the camera did not work as advertised*."

An example of the second category is "*not hesitate to*" in the sentence

"*I would not hesitate to recommend this product*,"

which is equivalent to the sentence

"*I would definitely recommend this product*."

A complete list of these preprocessing rules is included below.

Manually Discovered VS Patterns:

not(be)?(a)? disappoint(ed|ment)? → notdisappoint

not [article] (problem|complaint|issue) → noproblem

not (be|been|have|had) (a|any) (problem|complaint|issue|trouble|hassle) → noproblem

not a (good|great|bad|very ____) → not(good|great|bad|very ____)

not as ____ as → not____ compared to

not as ____ → not____

not the best → notgood

not the most ____ → not____

not [augmenter] [adj] → not[adj]

not [adj] → not[adj]

no ____ (problem|complaint|issue|trouble|hassle) → noproblem

no (problem|complaint|issue|trouble|hassle) → noproblem

Preprocessing rules:

Negation identification:

failed to → did not

(0 |0%|zero) → no

(problem|complaint|issue|trouble|hassle) free → noproblem

without [article]? (problem|complaint|issue|trouble|hassle) → noproblem

Reduced-scope negation:

if not the ____ → the ____

(not|no|without|never) hesitat(ed|e|ion|ing) (to)? → definiately

not recommend (____)? (____)? (____)? enough → recommend (____) (____) (____)

nothing (but|short of) → just

not (seem|appear)(ed|ing|s)? (to be|to|like|that)? (a)? (very)? → not

not (find(ing)?|found|notic(ed|e|ing)) it (to be)? → not

not (find(ing)?|found|notic(ed|e|ing)) (a|any|anything|one) → no

8 Results of Negation-Handling Approaches

Performance is calculated using the error categories described in the introduction (improper class (IC) and jump to judgment (JJ)), as well as the standard recall metric. Lost opportunity (LO) errors are accounted for by recall. Since the priority of this study is to minimize classification errors, especially IC errors, a relatively low recall figure is considered acceptable for all classifiers. Results for all approaches discussed in the previous section are included. Time efficiency is calculated by measuring the number of sentences classified per second.

To review the approaches from Section 7, Custom Windows with Preprocessing uses a handful of preprocessing rules to make negations more apparent, and then tags each word following a negation word as negated. The number of words tagged in this way is fixed for each negation word. Fixed-Window with Preprocessing uses a similar system, but with the same window size for all negation words. The plain Fixed-Window approach does not use the preprocessing rules. The POS-Based Heuristics approach also tags words following a negation word as negated, but uses heuristics based on part-of-speech information to determine the scope of each negation. Chunking uses an NLP-based shallow parser to determine negation scope. Manually Discovered Patterns use a completely different approach, relying on manually created syntactic rules for identifying and interpreting negations. Finally, the Base classifier does not use any negation-handling strategy.

Table 4 shows the 10-fold cross-validated results of all approaches with threshold parameters set to keep recall constant at approximately 62% (this recall value was chosen arbitrarily). The Custom Window with Preprocessing approach achieves the best performance, reducing IC errors by 28.72%. This improvement is statistically significant at $p < 0.001$ (all significances reported in this section were found by using the McNemar test). The Fixed-Window with Preprocessing approach performs second best, with a reduction in IC errors of 26.6%. Fixed Window tagging without Preprocessing achieves the next best performance, with a 23.05% reduction in IC errors. The manually discovered patterns reduce IC errors by 19.15%. The POS-Based Heuristic approach achieves improvement on IC errors of 17.02%. Chunking

resulted in the most modest performance improvement, with a reduction in IC errors of 4.61%.

Table 4. Results of all approaches on the 10-fold cross-validated set. Recall is held constant across all approaches at approximately 62%.

10-Fold Cross-Validation	IC	JJ	Recall		% IC Improvement
Base	2.82%	9.59%	62.03%		0.00%
Manually Discovered Patterns	2.28%	8.72%	62.06%		19.15%
Chunking	2.69%	9.20%	62.01%		4.61%
POS-Based Heuristics	2.34%	8.64%	62.01%		17.02%
Fixed-Window	2.17%	8.65%	62.01%		23.05%
Fixed-Window w/ Preprocessing	2.07%	8.68%	62.02%		26.60%
Custom Window w/ Preprocessing	2.01%	8.66%	61.99%		28.72%

Tests were also performed on the cross-validated set to test the techniques of ending negation scope at punctuation and using a single negation tag for all negations, as described previously. Both of these techniques reduced the system's performance. The results are not shown here.

Ending at punctuation was also tried (as in [Pang et al.]), but reduced performance when combined with the fixed-window technique. Another experiment compared the effects of using a single tag for all negations versus distinct tags for each negation word (e.g. "not," "no," and "never"), finding that distinct tags for each negation word performed better.

Table 5 shows the results of the various approaches on the set of all VS sentences. The approaches rank in similar order to their performance on the cross-validated set, though here the Fixed-Window w/ Preprocessing approach slightly outperforms the Custom-Window w/ Preprocessing approach. Another exception is that the Manually Discovered Patterns outperform Chunking and the POS-Based Heuristics. Fixed-Window w/ Preprocessing improves IC errors by 65.37%. This result is statistically significant at $p < 0.01$. The window-based tagging approaches also all improve recall in VS sentences by over 10%. Results in this table are not statistically significant due to the small number of data points.

Table 5. Results of all approaches on the set of all VS sentences.

All VS Sentences	IC	JJ	Recall		% IC Improvement
Base	9.76%	10.73%	46.18%		0.00%
Manually Discovered Patterns	6.15%	11.28%	45.61%		36.99%
Chunking	6.60%	10.38%	49.86%		32.38%
POS-Based Heuristics	7.14%	9.38%	52.97%		26.84%
Fixed-Window	4.72%	8.96%	51.84%		51.64%
Fixed-Window w/ Preprocessing	3.38%	9.66%	50.99%		65.37%
Custom Window w/ Preprocessing	3.85%	9.62%	50.99%		60.55%

Table 6 shows the results of all approaches on the set of sentences that do not contain VS. The four approaches that performed the best above provide essentially identical performance on this set, with a reduction in IC errors of 46.2%, though the results in this table are not statistically significant ($p = 0.18$). It may at first appear surprising that this study's negation-handling techniques so significantly reduced the incidence of IC errors on the set of sentences that *did not contain* negation. The reason for this change is that the enhanced classifier was better able to interpret the negation found in the training data, and thus it was able to perform better even on test data that did not include negation.

Table 6. Results of all approaches on the set of non-VS sentences.

Non-VS	IC	JJ	Recall		% IC Improvement
Base	1.71%	6.73%	65.85%		0.00%
Manually Discovered Patterns	1.62%	6.65%	65.85%		5.26%
Chunking	1.19%	6.84%	65.14%		30.41%
POS-Based Heuristics	0.92%	6.82%	64.62%		46.20%
Fixed-window	0.92%	6.64%	64.69%		46.20%
Fixed-Window w/ Preprocessing	0.92%	6.99%	64.62%		46.20%
Custom Window w/ Preprocessing	0.92%	6.98%	64.75%		46.20%

Tables 7-10 show the results of the best-performing approach (Custom Window with Preprocessing) on the four categories of VS. The strongest improvement is shown in the category of Negated ADJPs, with a reduction in IC errors of 88.93% and an improvement in recall of 79.99% (Table 9). Significant

improvements in Negated VPs and Negated NPs are also evident in all three metrics (Tables 7 and 8). This approach did not improve the system's performance on Modal phrases, and actually reduced precision and recall sharply, though it should be noted that these phrases amount to a very small number of data points. It can also be noted that the classifier that included the negation handling approach did not actually produce more errors in the Modal class, but rather its true positive rate was much lower, leading to a higher proportion of IC errors. Due to the small number of data points, only the results in tables 7 and 9 are statistically significant (at $p < 0.05$ and $p < 0.01$, respectively).

Anecdotally it can be said that many of the IC errors that remain in the categories of Negated VPs and Negated NPs are not due to negation itself by rather to other phenomena. A primary example is a statement that communicates positive sentiment about the item at hand by comparing it with a different item, which is described as very negative. An example would be the following: "No problems with this camera, unlike my last one which broke after two weeks and couldn't even be returned for a refund." This study does not address this issue.

Table 7. Results of the Custom Window with Preprocessing approach on negated verb phrases.

Negated VP	IC	JJ	Recall		% IC improvement
Base	8.62%	6.90%	53.55%		
Custom Window w/ Preprocessing	4.50%	6.31%	54.10%		47.80%

Table 8. Results of the Custom Window with Preprocessing approach on negated noun phrases.

Negated NP	IC	JJ	Recall		% IC improvement
Base	5.45%	9.09%	48.45%		
Custom Window w/ Preprocessing	1.75%	8.77%	52.58%		67.89%

Table 9. Results of the Custom Window with Preprocessing approach on negated adjective phrases.

Negated ADJP	IC	JJ	Recall		% IC improvement
Base	21.05%	26.32%	32.79%		
Custom Window w/ Preprocessing	2.33%	13.95%	59.02%		88.93%

Table 10. Results of the Custom Window with Preprocessing approach on modal phrases.

Modals	IC	JJ	Recall		% IC improvement
Base	18.18%	27.27%	24.00%		
Custom Window w/ Preprocessing	20.00%	40.00%	16.00%		-10.01%

Table 11 shows the results of all approaches on the timing test. The system was not optimized for speed and the results are only for comparison with each other. The system enhanced with the Custom Window with Preprocessing approach is able to classify at 80% of the speed of the base classifier. The Manually Discovered Patterns and Fixed Window without Preprocessing approaches ran at approximately the same speed, 96% as fast as the baseline. The Chunking and POS-Based Heuristics approaches were significantly slower. Chunking runs at only 9% of the speed of the base classifier, while the POS-Based Heuristic approach runs at 32% of the baseline speed.

Table 11. Results of the speed test for all approaches.

Timing Test	Sentences/Second
Base	770
Manually Discovered Patterns	740
Chunking	72
POS-Based Heuristics	250
Fixed Window	740
Fixed Window w/ Preprocessing	620
Custom Window w/ Preprocessing	620

9 Discussion of Results

The approach found to perform best by this study is similar to that used in previous works (such as [Pang et al.], [Dave et al.], and [Kennedy and Inkpen]), though the results reported by this study are very different than those reported by others. While others reported a minor improvement (less than 3% in [Kennedy and Inkpen]) or even a slight worsening of performance from using this technique [Dave et al.], this study shows a major benefit. This difference can be explained by important differences between the technique and evaluation of this study and those of others. The primary difference in the technique is the method of determining the scope of a negation. The difference in evaluation is this study's separation of the errors into the distinct categories of "Improper Class" (IC) and "Jump to Judgment" (JJ).

The method of determining negation scope found by this study to be most effective (Custom-Sized Windows) was not used by previous works. As reported in the Related Work section, Pang et al. consider all words that come before the next punctuation as within the scope of the negation, Dave et al. appear to consider all words within the rest of the sentence, and Kennedy and Inkpen perform shallow parsing to determine the proper scope. Our results using scoping methods similar to those used by other works are in line with what those works' authors report. For example, this study's experiments using shallow parsing ("Chunking") show performance only slightly above that of the baseline classifier, and stopping the scope only at punctuation or at the sentence's end also performed poorly.

As described previously, this study considers it important to separate sentiment classification errors into distinct classes, IC, JJ, and LO. This study's most significant results come in the infrequent but highly costly error category of IC errors. However, if these error classes were combined, this study's results would appear to show a minor benefit. On the ten-fold cross-validation results, conflating IC and JJ errors into one precision metric would show a maximum change in precision from 88.74% to 89.33%, an improvement of only 0.66%. Even on the set of VS sentences

this merged precision metric would only go from 82.5% to 85.84%, yielding a 4.05% improvement.

The study by [Mutalik et al.] that addressed negation was performed in a significantly different application domain, and thus its techniques are not applicable to sentiment classification. That work aimed to improve medical concept detection for indexing free-text medical documents by identifying concepts whose presence was negated. For example, a document should not be indexed as containing the concept "fracture" if the only occurrence of that concept is negated, as in the phrase "no evidence of fracture." The semantic interpretation of negation in concept detection is significantly different than the semantic interpretation of negation in sentiment classification. In concept detection, negation is compositional; that is, presence of a concept, when negated, becomes the logical opposite, absence of that concept.

This compositionality does not hold in sentiment classification. Intuitively, "great" is extremely good while "not great" is only modestly bad. Conditional probabilities from the sentiment classification model help to illustrate this point. If negation were compositional in sentiment, we would expect that if a word is predictive of positive sentiment with a probability of X, then when negated it would be predictive of negative sentiment with the same probability. This can be shown to be incorrect with the class conditional probabilities learned by the sentiment classification model shown in Table 12.

Table 12. Class-conditional probabilities for words and their negated counterparts. Since words and their negated counterparts do not have the same class-conditional probabilities for opposite sentiments, negation of sentiment is not compositional.

Positive Words			
Word	P(pos)	NegatedWord	P(neg)
good	0.88	notgood	0.96
work	0.67	notwork	0.96
great	0.98	notgreat	0.86
fast	0.92	notfast	0.2
Negative Words			
Word	P(neg)	NegatedWord	P(pos)
fail	0.84	notfail	0.68
return	0.84	notreturn	0.19
break	0.86	notbreak	0.68

10 Conclusion and Future Work

This study has examined the impact of valence shifters in sentiment classification. Valence shifters, with negation as the most common example, have been shown to be both common and problematic for current state-of-the-art sentiment-classification systems. This study described an approach that yields significant success in handling negation. This approach tags words closely following a negation as being negated, replacing the original word with the concatenation of the negation word and the original word. The most effective variant of this approach tagged all words within a window that has a fixed size specific to each negation word.

Remaining errors involving negation are mostly due to other phenomena, such as a positive statement that references a negative quality of a different item. Modals have not been addressed, and are indeed quite problematic for the classifier used in this study, though they are very uncommon.

This study has uncovered some important areas for future work that could further improve sentiment classification. The first is a technique for identifying if the subject of a semantically-charged phrase is the principal subject of the review, or if it is instead a different item, which may be described negatively or positive in order to contrast the subject at hand.

A second important area for future work in sentiment classification would be a method of automatically identifying words and phrases that cause negation. Though this study relied on a manually-created list of negators, including words like “not,” “no,” and “never,” this list was insufficient to capture many of the negations actually found in the corpus. Some examples of other phrases that cause negation are “failed to,” and “hard … to,” as in the sentence

> “*The overly detailed approach makes it a hard book to recommend enthusiastically*.”

A related target for future work would be a method of automatically identifying words and phrases that neutralize a negation. These would include phrases

like “hesitate to.” The role of such phrases can be seen by contrasting the sentiment of the following two sentences:

“*I would not want to recommend this to anyone*,”

and

“*I would not hesitate to recommend this to anyone*.”

REFERENCES

K. Dave, S. Lawrence, and D. M. Pennock. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proc. of the 12th Int.WWW Conf.*, 2003.

T. Dunning. Accurate Methods for the Statistics of Surprise and Coincidence. *Computational Linguistics*, 19(1): 61-74.

M. Gamon, A. Aue, S. Corston-Oliver, E. Ringger. Pulse: Mining Customer Opinions from Free Text. In *Lecture Notes in Computer Science*. 3646: 121-132, 2005. Springer Verlag.

A. Kennedy and D. Inkpen. Sentiment Classification of Movie and Product Reviews Using Contextual Valence Shifters. *Computational Intelligence*, 22(2), 2006.

L. Lee. A Matter of Opinion: Sentiment Analysis and Business Intelligence. IBM Faculty Summit on the Architecture of On-Demand Business. 2004.

P. G. Mutalik, A. Deshpande, P. M. Nadkarni. Use of General-purpose Negation Detection to Augment Concept Indexing of Medical Documents. *Journal of the American Medical Informatics Association*. 2001 Nov–Dec; 8(6): 598–609

T. Nasukawa and J. Yi. Sentiment Analysis: Capturing Favorability Using Natural Language Processing. In Proceedings: *The Second International Conference on Knowledge Capture (K-CAP)*, 2003.

B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up? Sentiment classification using machine learning techniques. In *Proc. of the 2002 ACL EMNLP Conf.*, 2002: 79–86

L. Polanyi and A. Zaenen. Contextual valence shifters. In J. Shanahan, Y. Qu, and J. Wiebe (eds.), Computing Attitude and Affect in Text: Theory and Applications. 20: 1-9, 2004. Springer-Verlag.

M. Porter. An Algorithm for Suffix Stripping. *Program*, 1980, 14(3): 130−137

M. Rimon. Sentiment Classification: Linguistic and Non-Linguistic Issues. In *Proceedings of Israel Association for Theoretical Linguistics 21*. 2005. http://micro5.mscc.huji.ac.il/~english/IATL/21/. Retrieved 11/2006.

P. D. Turney. Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. In *Proc. of the 40th ACL Conference*, 2002: 417–424

V. N. Vapnik. The Nature of Statistical Learning Theory. 1995. Springer.

C. White. Consolidating, Accessing and Analyzing Unstructured Data. Business Intelligence Network, 2005. http://www.b-eye-network.com/view/2098. Retreived 11/2006.

J. Yi et al. Sentiment Analyzer: Extracting Sentiments about a Given Topic using Natural Language Processing Techniques. *Third IEEE International Conference on Data Mining*, 2003: 427- 434