A LINGUISTIC MODEL FOR IMPROVING SENTIMENT ANALYSIS SYSTEMS

A Thesis
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By
Jared Coleman Hall

In Partial Fulfillment of the Requirements
for the Degree of
MASTER OF SCIENCE

Major Department:
Computer Science

March 2014

Fargo, North Dakota

North Dakota State University
Graduate School

Title

A Linguistic Model for Improving Sentiment Analysis Systems

By

Jared Coleman Hall

The Supervisory Committee certifies that this ***disquisition*** complies with North Dakota State University's regulations and meets the accepted standards for the degree of

MASTER OF SCIENCE

SUPERVISORY COMMITTEE:

Dr. Simone Ludwig
Chair

Dr. Wei Jin

Dr. Stephenson Beck

Approved:

3/24/2014
Date

Dr. Brian M. Slator
Department Chair

ABSTRACT

The value of automated sentiment analysis systems is increasing with the vast amount of consumer-generated content, allowing researchers to analyze the information readily available on the World Wide Web. Much research has been done in the field of sentiment analysis, which has improved the accuracy of sentiment analysis systems. But sentiment analysis is a challenging problem, and there are many potential areas for improvement. In this thesis, we analyze two linguistic rules, and propose algorithms for these rules to be applied in sentiment analysis systems. The first rule is regarding how a sentiment analysis system can recognize and apply the semantic orientation of opinion headings in product reviews to features discussed in the review. The second rule we propose allows the sentiment analysis system to recognize informal forms of words used in analyzed documents. Additionally, we analyze the effects of spelling mistakes in text being analyzed by sentiment analysis systems.

ACKNOWLEDGEMENTS

I am grateful to those who made this thesis possible. First and foremost, I would like to thank Dr. Simone Ludwig, whose guidance, direction, and mentorship through research process, and the process of writing the thesis was invaluable. Also, many thanks to Dr. Wei Jin for her instruction in sentiment analysis, and her recommendations in topics of research in this area. Her help became an important part of this thesis.

DEDICATION

For my family

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

Algorithm Page

1. INTRODUCTION

This chapter provides an introduction to sentiment analysis and the motivation behind this research project. We also provide an introduction to the contribution this thesis provides in the sentiment analysis field. Finally, the chapter provides an outline of the thesis.

1.1. Introduction to Sentiment Analysis

Both consumers and businesses can gain value from knowing the opinions of others with respect to a product. Consumers use the opinions of other consumers to make purchasing decisions, while businesses use market research as a method to determine what consumers really want, not just what they think consumers want. Traditional methods of market research include opinion polls, surveys, focus groups, and personal interviews, which require the researcher to solicit and gather information from the consumer. [1]

The World Wide Web has transformed the way in which people express their views and opinions. With the explosion of e-commerce, blogging, online forums and social media, vast amounts of information related to consumer sentiment are readily available to researchers [2].

Automated opinion mining, or sentiment analysis, is a method used to determine attitudes and opinions with respect to a topic, and is a challenging natural language processing, or text mining, problem [3]. With this method, automated systems can supply summarized views of information based on the vast amounts of consumer sentiment data expressed, and made publicly available, on the World Wide Web.

Early research into sentiment analysis focused on determining an overall sentiment orientation for each review. For example, Turney [4], in research in 2002, aimed to classify product reviews as recommended (thumbs up), or not recommended (thumbs down). He focused his research on reviews from four product categories: automobiles, banks, movies, and travel destinations. For each review, the semantic orientation of the review is determined by summarizing the orientation of the opinion phrases contained within the review. Turney proposed in his research two algorithms for determining the orientation of phrases with calculations including comparisons of words in the phrase to words with a known orientation. [4]

More recent research has looked into the more granular level of opinion mining of feature-based sentiment analysis. This form of sentiment analysis, rather than looking at the sentiment for the review or document as a whole, categorizes the sentiment by identifying product features on which the document or review expresses an opinion [5]. For example, if the document is a product review of a camera, the goal of the sentiment analysis is to find each feature of the camera contained in the review, such as lens, flash, picture quality. After identifying features of the object in the review, the goal is to determine the sentiment orientation the reviewer holds for each of these features. The focus of this thesis is on feature-based sentiment analysis.

1.2. Motivation

The motivation for this research is to assist in improving the overall effectiveness of sentiment analysis systems. A lot of research has been done in the area of sentiment analysis. Much of the related work has been successful in expanding the accuracy and application of sentiment analysis. However, as noted by Ogneva, no system will ever be as accurate as human analysis. There are subtleties in the language, such as sarcasm, for which a computer is not able to account [6].

Another challenge is that language continues to evolve. The vocabulary, methods, and linguistic patterns employed in user-generated online content changes as new technologies become available. To remain accurate and relevant, a sentiment analysis system must evolve with the language. For example, Jiang, et al. [7], had trouble in their research classifying the Twitter message stating "#lakers b**tch!" (noting that the expletive was spelled out fully in the original message). Using language that would traditionally be considered negative in semantic orientation, it is a language subtlety to understand that, within the context in which it was used, the word provides a positive semantic orientation towards the Lakers. With the changes that occur, and the complexity of language overall, our goal and motivation is to help drive some of the evolution that will help improve sentiment analysis.

1.3. Contribution

The basis for the majority of the work in this thesis comes from one research paper in particular: *A Holistic Lexicon-Based Approach to Opinion Mining* by Ding, Liu, and Yu [8]. The paper proposes a model for feature-based sentiment analysis with many linguistic rules that provide a good foundation for the additional rules proposed in this thesis.

The contribution of this thesis is to propose additional linguistic rules that improve the performance of feature-based sentiment analysis systems that focus on online product reviews. The first contribution of this research is to propose a linguistic rule which will improve the accuracy with which a sentiment analysis system assesses the semantic orientation of product features. Many individuals who write online product reviews will write the review in a format that categorizes the feature comments by orientation, providing a heading to the section that indicates the orientation of the features in the section. For example, there may be a section in the review for “Pros” and a section for “Cons”. The goal of the proposed rule is to identify opinion headings used within a review, identify the features contained within the section associated with the opinion heading, and apply the semantic orientation of the heading to the features within the section.

The second contribution of this paper is to propose rules that understand informal forms of words that may be viewed as spelling mistakes. Informal forms of words, such as “mic” as a form of “microphone”, may not be recognized as words in the dictionary. This can affect how the system understands the word, and affect the semantic orientation scores assigned by the sentiment analysis. This thesis proposes some simple rules for identifying informal forms of words that can be applied to the analysis process.

The third contribution of this research is regarding the role that correct spelling has within the effectiveness of sentiment analysis systems. The thesis does not propose automated methods for correcting all spelling mistakes that can occur within a review. The goal is simply to understand the impact that spelling correction has on the effectiveness of the system.

For this research we built a software system, called Sentience. This system implements the opinion mining rules and conventions discussed in [8], as well as the new rules proposed as contributions for this thesis.

1.4. Outline of Thesis

In Chapter 2 of this thesis, we provide an overview of previous work and methods in feature-based sentiment analysis, focused primarily on the research in [8]. Chapter 3 provides the details of the implementation of the Sentience system. The section provides details on the rules and algorithms used for the sentiment analysis. In Chapter 4, we provide the assessment of the effectiveness of the system through empirical evaluation. We then summarize our conclusions for the research in Chapter 5, and discuss potential future work in sentiment analysis.

2. METHODS OF SENTIMENT ANALYSIS

In this chapter, we provide definitions related to the sentiment analysis that will be used throughout the thesis, as well as summary of prior work in the field of study.

2.1. Definitions

Sentiment analysis is the analysis of people's opinions, sentiments, evaluations, appraisals, attitudes, and emotions towards entities such as products, services, organizations, individuals, issues, events, topics, and their attributes [9]. The following are definitions of terminology related to sentiment analysis, as defined in [8], that are used in this thesis:

- **Semantic orientation of an opinion:** The semantic orientation of an opinion on a feature *f* states whether the opinion is positive, negative, or neutral.
- **Opinion holder:** The *holder* of a particular opinion is the person or the organization that holds the opinion. In this thesis, as our research focuses primarily on product reviews, the term *reviewer* is used interchangeably with *opinion holder*.
- **Object:** An *object O* is an entity which can be a product, person, event, organization, or topic.
- **Feature:** A *feature* of an object is a characteristic or component of the object.

In addition to the above definitions identified in prior research, we use the following definitions in this thesis:

- **Review:** In this thesis, we use the term *review* to identify a document, written by an opinion holder, expressing an opinion about an object *O*. Sentiment analysis systems can be used to analyze many different document types.

However, we focus on product reviews for our research. The terms *review* and *document* are used interchangeably in the thesis.

- **Feature instance:** A *feature instance* is a single occurrence of a feature *f* within a review. For example, a camera has a feature "lens". If an opinion holder in a review of the camera mentions the camera lens four times, then we consider the review to have four feature instances of the "lens" feature.
- **Descriptor:** A *descriptor* is an opinion word used to describe a feature instance in a review. This term is used interchangeably with the term *opinion word* in this thesis.

2.2. Features and Opinions

The goal of feature-based sentiment analysis is two-fold; first, identify feature instances contained in a document on which the opinion holder of the document has commented; then determine the semantic orientation of the opinion held by the opinion holder for each feature [8].

One of the challenges of identifying features in a review is that features can be either explicit or implicit. If the feature appears in the review, then it is considered to be an *explicit feature*. If the feature is not explicitly identified in the review, but rather is implied, then the feature is considered to be an *implicit feature*. [8]

For example "picture quality" in the following sentence is an explicit feature:

"The picture quality of this camera is incredible."

The sentence below provides an example of an implicit feature. Although the sentence does not use the word "price", the feature is implied by the adjective used to describe the product.

“This television is just too expensive.”

Similarly, the opinions expressed for product features may also be either explicit or implicit. The sentence below provides an example of an explicit opinion, showing a positive semantic orientation of the opinion of the feature “screen resolution”:

“The screen resolution on this table is beautiful.”

The following sentence provides an example of a sentence with an implicit opinion expressed regarding the quality of the product:

“The radio broke after just two days.”

The semantic orientation of an opinion is typically measured on either a binary scale (positive or negative), or a ternary scale (positive, negative, or neutral) [10]. However, some research has been done in assigning a scale to the orientation to understand the degrees and strength of the sentiment [11]. Research has also been done to classify the orientation of the opinion holder’s feelings based on emotion, rather than on a scale. In their research, Denis, et al. [10], use sentiment analysis to classify opinions in online documents based on what Ekman identified as the six universal categories of emotion [12]: joy, fear, sadness, anger, disgust, and surprise. Rather than rating document orientation on a scale, the research in [10] attempts to display for the user the emotions that are felt towards the topic in each document.

2.3. Lexicon-Based Methods

Lexicons can be used as a method for determining the semantic orientation of opinions in a document. Most techniques for sentiment analysis use, to some degree, a lexicon of opinion-bearing words to understand the semantic orientation of opinions expressed in the text [8]. The approach uses a lexicon, or list of words, with a pre-defined semantic orientation. The semantic orientation for the lexicon for the approach taken in [8] is based on a ternary scale, including the options of positive (+1), negative (-1), or neutral (0).

Thelwall, et al. [11] attempted to identify the strength of an opinion, rather than limiting the analysis to the orientation of the opinion. In their work, they developed a lexicon in which each opinion word is given an orientation, as well as a strength score on a scale from 1 to 5. This score is then used in determining the strength of the opinion expressed in the document analyzed by the system. Their research focused on comments from the social media site MySpace (http://www.myspace.com). With the communication methods that are typical of social media, they included other emotional signals, such as emoticons, in their algorithm for determining the semantic orientation of a statement.

The complexity in the application of the lexicon varies by system. Some simpler implementations of sentiment analysis systems, such as the Analytics for Twitter 2013 program [13], simply search through the text of the document being analyzed for words existing in the lexicon, and assign an orientation score based on the orientation assigned in the lexicon. More sophisticated systems, such as the system proposed in [8], use linguistic rules to understand the context in which the lexical words are used, and assign an orientation score based on the rules.

2.4. Linguistic Rules

A significant amount of research has been done to identify linguistic rules and methods to improve the accuracy of sentiment analysis systems. The research in [8] proposed rules to solve two key problems. The first problem deals with context-dependent opinion words. Many prior methods of sentiment analysis did not have a mechanism for dealing with opinion words in a review where the sentiment orientation is dependent on the context in which it is used [8]. For example, the word “long” can indicate a positive or negative sentiment orientation depending on the product feature it is describing, and the context in which it is used. A *long* battery life in a review of a mobile phone may indicate a positive sentiment, while a camera taking a *long* time to focus would indicate a negative orientation.

To deal with this problem, Ding, et al. [8], propose a method for looking not only at the current sentence alone to determine the orientation, but also using external information and evidences in other sentences, and other reviews of the same product features, to determine the orientation of the current feature instance and descriptors. They propose several linguistic conventions in natural language expressions to infer the orientation of opinion words, and can then apply the orientation of an opinion word in other sentences and reviews for a defined product feature. The global nature of the analysis leads them to refer to the method as a “holistic” approach.

While the authors in their research in [8] provide methods for solving these problems using linguistic rules, other researchers have attempted to solve the problems of context-dependent orientation using an approach that more closely resembles a lexicon-based approach. These methods expand the concept of a lexicon to include not only the orientation of individual words, but also the orientation of combinations, or sets, of words. For example, WordNet is a lexical database that identifies relations among English words. The database has a list of English words that provides an explanation of different senses of the word when put into various combinations, or “synsets”, with other words in the list. [14]

For example, a synset in WordNet may be “cold beer”, in which the sense of the word “cold” provides the meaning “having a cold temperature”. However, if the word “cold” is used in the synset “cold person”, then it has the meaning “being emotionless”. [15]

Additional research has expanded the functionality for opinion mining with WordNet. One example is outlined in research by Esuli and Sebastiani [16]. In their work, they created SentiWordNet, in which they add to each WordNet synset a polarity score, giving a context-dependent orientation to words in the WordNet lexicon [17]. The SentiWordNet system assigns a polarity to each of these sets of words to assign an orientation to words in the context in which they are used [15].

Some systems have implemented SentiWordNet into their processes as a method for dealing with the issue of context-dependent orientation. For example, Guerini, et al. [15], implemented methods for deriving past polarities from SentiWordNet, and applying those polarities to the semantic orientation of the document analyzed by the system. Paramesha and Ravishankar [18] implemented SentiWordNet as a method for improving the accuracy of sentiment analysis in cross-domain product reviews, as different words may have a different meaning depending on the context or domain of the product being reviewed.

An additional extension of WordNet was created by Strapparava and Valitutti [19]. In their work they developed WordNet-Affect, which adds a new layer to the WordNet system that categorizes synsets by mental states, or affective labels. The categories of labels include emotion, mood, trait, cognitive state, physical state, edonic signal, emotional response, behavior, attitude, and sensation. Each synset is then assigned to these categories, and can then be used to provide greater insight in natural language processing.

SenticNet 2 is a lexical resource similar to SentiWordNet and WordNet-Affect. It is a semantic and affective resource that assigns a polarity value to about 5,700 concepts, and assigns cognitive and affective information to about 14,000 concepts. The categorized concepts in the SenticNet 2 system are similar to the synsets in WordNet. This polarity and affective information can then be extracted for performing sentiment analysis, notably in determining the orientation of context-dependent words. [20]

The second major problem the authors aim to solve in their research in [8] is related to situations where there are multiple, conflicting opinion words in the same sentence. Opinion words in the same sentence as a product feature are assumed to have an association with the product feature. If there are multiple, conflicting opinion words in the same sentence, prior lexicon-based approaches were unable to effectively determine which opinion word should be used to determine the sentiment orientation of the opinion held by the writer of the review, relative to the product. [8]

To deal with this problem, the authors propose a method to aggregate the orientation of conflicting opinion words by considering the distance between the opinion word and the product feature. The farther an opinion word is from a product feature, the less weight it is given in determining the semantic orientation relative to the product feature. [8]

Although Ding, et al. [8], find this method to be highly effective in calculating orientation scores for each feature, Mukherjee and Bhattacharyya [21] say this, and similar methods, can be improved. In their research they point to specific scenarios in which a sentence will contain multiple feature instances and distributed emotions, and the feature descriptors closest to the feature instances are not necessarily related to the instance. They propose linguistic rules for determining the relation between opinion words and feature instances within a sentence. In their paper they provide a proposed algorithm for calculating a dependency relation among opinion words and feature instances.

2.5. Additional Research

The rules and algorithms used in this thesis are based primarily on the research done in [8], but many other researchers have provided work on this topic, proposing linguistic rules for sentiment analysis. Ganapathibhotla and Liu [22] proposed rules for mining opinions from sentences that compare two products, rather than making a statement about a single product. For example, a reviewer may state:

"The picture of Television X is much sharper than the picture of Television Y."

After identifying the objects and object features in the sentence, the proposed algorithm then determines which, of the two, is the preferred entity. [22]

Zhai, et al. [23], propose algorithms for clustering product features in opinion mining. One of the tasks of feature-based sentiment analysis is identifying the product features discussed by the opinion holder in the document. Many systems, including the system designed for the research in [8], do not do grouping or categorization of product features. Thus, the resulting output from the system lists each feature instance separately, with an orientation score that must be analyzed individually. The research performed by Zhai, et al. [23], provides guidance for grouping feature instances that reference the same product feature. The proposed method includes grouping words by shared words (words that exist across feature instances), and feature instances with lexical similarities. The system output, then, is a list of product features with an orientation score that is a summary of the associated feature instances. [23]

Tan, et al. [7], attempted to improve the accuracy of sentiment analysis in social media by taking advantage of the social relationships that are present on the social media sites. The primary concept behind their research is the tendency for individuals to have similar opinions as those with whom they have close personal relationships, or "birds of a feather flock together". They propose algorithms for determining sentiment orientation of comments by a social media user by incorporating information from other users with whom the user has close links.

The experiment for the research was performed against data from Twitter. They determined relationships among users by reviewing the user's followers, as well as who the user is following. They also incorporated into the equation an analysis of other users addressed in users' comments using the Twitter @-convention. Using these relationships, the authors found that sentiment analysis can be improved significantly by incorporating information from relationships on the social media site. Pulling data from related users, and applying orientation information from the related users' content provides context, and improves the analysis. [7]

Research has also been done with sentiment analysis as a method to predict future product sales. Archak, et al. [24], created a pricing model for optimizing future sales based on analyzed sentiment of product features in online product reviews.

Although these concepts and rules, as well as proposals in other research, appear to be effective in improving accuracy of sentiment analysis systems, and expanding the business use cases for sentiment analysis, we did not implement these additional rules as part of the basis for our research in the program for this thesis. Our focus was based on the rules applied from the research in [8].

3. THE SENTIENCE SYSTEM

To test the proposed rules for the thesis, we developed a software program called Sentience. In this chapter, we provide a detailed look at the functionality of Sentience, including the implementation of the rules and algorithms for solving the sentiment analysis problems.

Figure 1 provides a high-level overview of the processes implemented in Sentience for performing the feature-based sentiment analysis. The system takes product reviews as input into the system. The first step in the process for each review is to determine the part of speech of each word in the review. The system then parses through each review to separate the text into paragraphs and individual words. The third step is to identify each feature instance contained in the review. Fourth, the system identifies the descriptor words associated with each feature instance. After identifying the descriptors for each feature instance, the system attempts to identify the semantic orientation of the opinion held for each feature instance. To begin this, as the fifth step of the overall process, the system determines the word orientation of each descriptor using a lexicon. The next step is to apply several linguistic rules to further refine the semantic orientation score for each feature instance. The last step in the process is then to calculate the semantic orientation per feature in the review. Figure 1 also shows that the WordNet lexicon is used as part of the process in three of these steps. These steps are described in detail in this chapter.

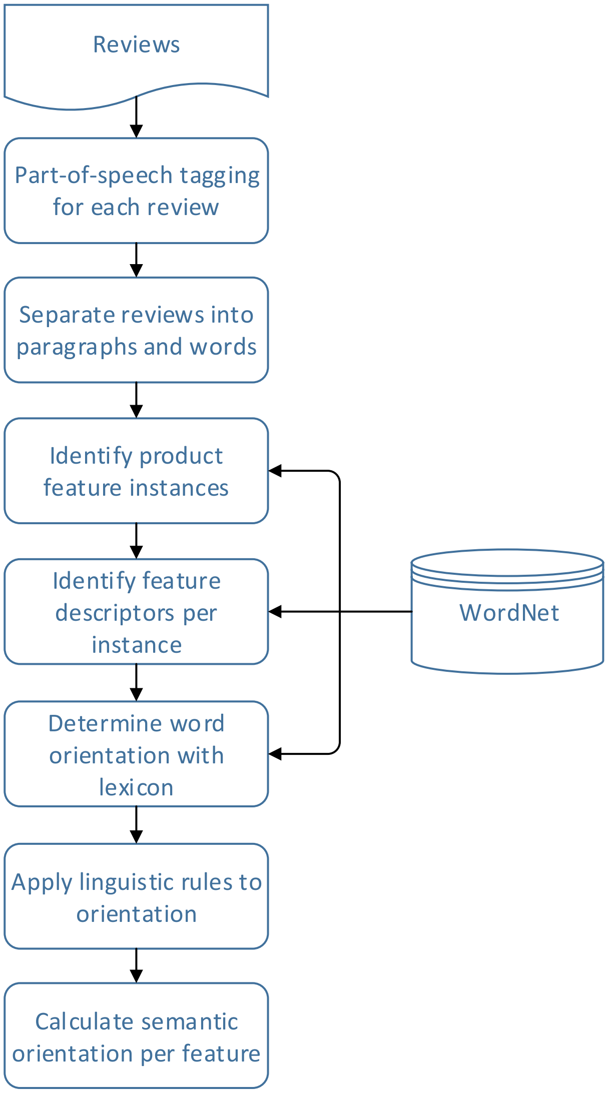

Reviews
Part-of-speech tagging for each review
Separate reviews into paragraphs and words
Identify product feature instances
Identify feature descriptors per instance
Determine word orientation with lexicon
Apply linguistic rules to orientation
Calculate semantic orientation per feature
WordNet

Figure 1. Sentience flow chart

Figure 2 provides a class diagram of the system. The figure shows the attributes and operations of each class in the system, as well as the relationships among the classes.

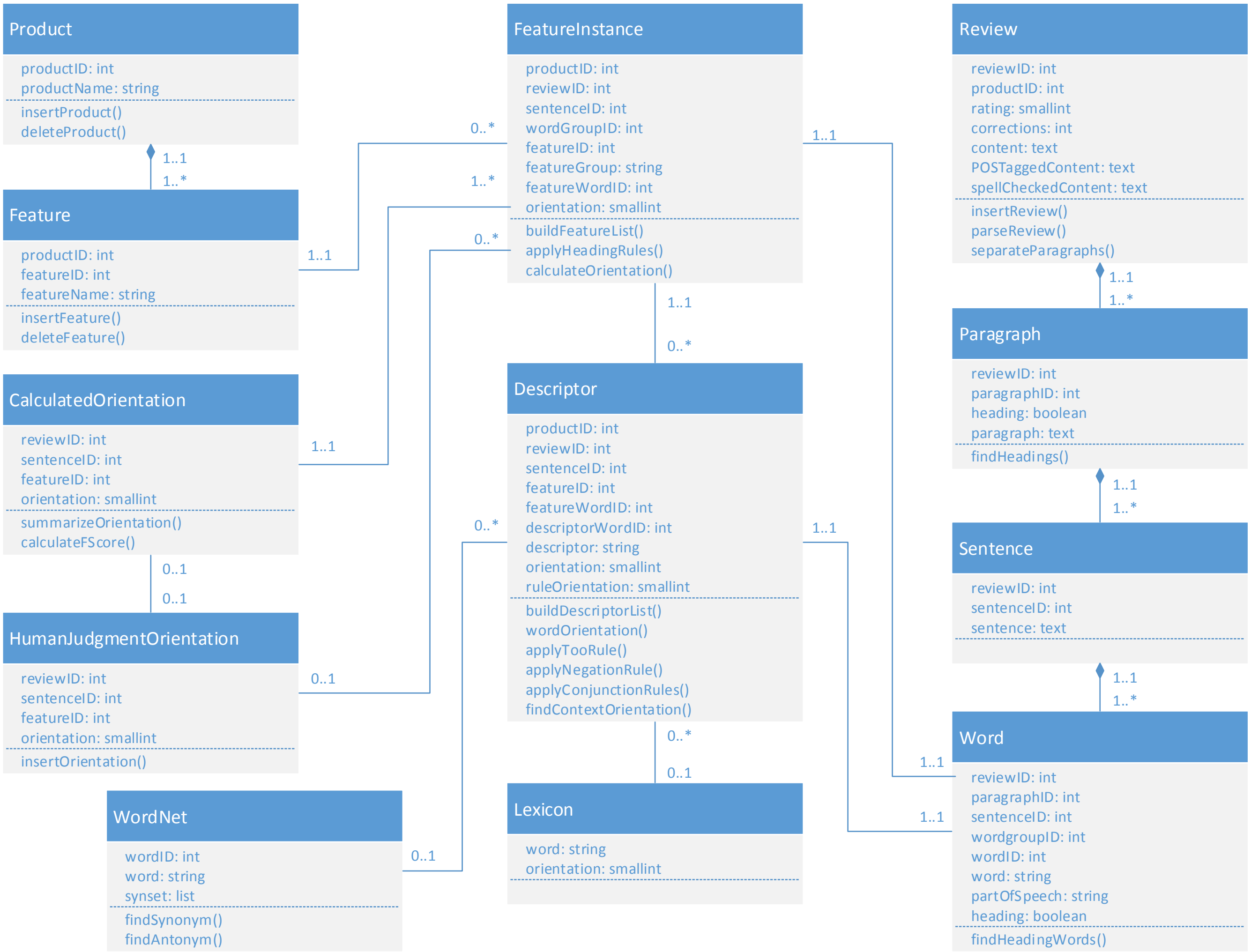


Figure 2. Sentience class diagram

3.1. System Input

Product reviews are the input into the Sentience system. These reviews can be taken from ecommerce sites, such as Amazon.com, BestBuy.com, or any site that provides consumers the ability to write product reviews. For this version of Sentience, the system does not automatically connect to these sites to gather the review information. The reviews must be copied and inserted into the system. The full text of the review, and product's overall rating for the review, are inserted as inputs into the system. For reviews on

Amazon.com, the product's overall rating is provided as a number of stars on a five-star scale.

3.1.1. NLProcessor

The system input also includes the text of the review that has been tagged with parts of speech. This is done by running the NLProcessor against each review [25]. Using the NLProcessor for part-of-speech tagging is a process used in the research for [8]. Any text can be used as an input into the NLProcessor system, and the system output is the text with each word and word group separated and tagged with part-of-speech information.

For example, consider the following statement from a product review that can be used as input into the NLProcessor:

"These shoes provide more foot support than any pair of running shoes I have ever owned."

The output of the NLProcessor would then be the following tagged text:

([These_**DT** shoes_**NNS**])

<: provide_**VBP :>**

([more_**JJR** foot_**NN** support_**NN**]) than_**IN** ([any_**DT** pair_**NN**]) of_**IN** ([running_**VBG** shoes_**NNS**]) ([I_**PRP**])

<: have_**VBP** ever_**RB** owned_**VBN :>** ._.

The text for each review with part-of-speech tagging is an additional input into the Sentience system. The part-of-speech information for each word can then be used in other procedures of the sentiment analysis process, as outlined in later sub-sections of this chapter of the thesis.

3.1.2. Word and Paragraph Parser

With the text of each review, and text for the review with part-of-speech information input into the system, Sentience then has to parse through the text to put the data into a format in which it can be understood and analyzed by the system. This is done by separating the text into paragraphs and individual words.

The process for separating each review into paragraphs simply steps through the text of each review looking for carriage returns in the text. For each carriage return, the text between the carriage returns is inserted into a table as a paragraph, with a paragraph ID.

The process for separating the text with part-of-speech tagging into individual words is more complicated than that of separating paragraphs. The procedure steps through the tagged review, and must recognize words, word groups, punctuation, special characters, and tagging information so it can all be separated out into the ParsedWords database table for analysis. For example, the tagged text shown in the example in section 3.1.1 would be parsed by the system procedures, and inserted into the ParsedWords table, as shown in Table 1 below.

Table 1. Format of the ParsedWords Database Table

REVIEWID	PARAGRAPHID	SENTENCEID	WORDGROUPID	WORDID	WORD	Part_of_Speech
1	1	1	1	1	These	DT
1	1	1	1	2	shoes	NNS
1	1	1	2	3	provide	VBP
1	1	1	3	4	more	JJR
1	1	1	3	5	foot	NN
1	1	1	3	6	support	NN
1	1	1	4	7	than	IN
1	1	1	5	8	any	DT
1	1	1	5	9	pair	NN
1	1	1	6	10	of	IN
1	1	1	7	11	running	VBG
1	1	1	7	12	shoes	NNS
1	1	1	8	13	I	PRP
1	1	1	9	14	have	VBP
1	1	1	9	15	ever	RB
1	1	1	9	16	owned	VBN

3.2. Identifying Product Feature Instances

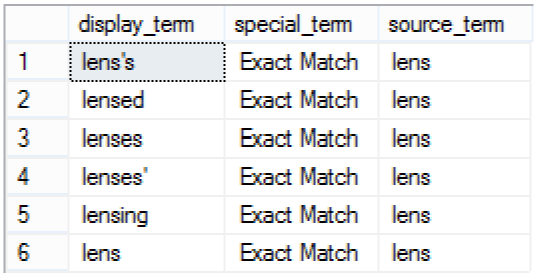
The Sentience system allows the user to define a list of features for which the system will search in each review. The user enters the list of features for which they want to see an analysis, and Sentience then takes that input, and searches through the reviews for instances of the listed features in each review.

For the Sentience system, we used a relatively simple approach for identifying the list of feature instances discussed in each review. We based our method for finding the instances on the assumption that the majority of features identified by reviewers, and entered as input, will be explicit features, primarily identified using nouns in the review.

An aspect of Sentience that improved its effectiveness in finding feature instances was the implementation of code that utilizes SQL Server's built-in Full-Text Search functionality. Full-Text Search provides the ability to perform linguistic searches against character or text fields in the database. The queries are based on linguistic rules for words and phrases of the language set for the data. [26]

The three Full-Text Search functions used in the Sentience code are sys.dm_fts_parser, FORMSOF [27], and FREETEXT [28]. These three functions allow the user to find a list of the various forms of a word or list of words. For example, the following query provides a list of the various forms of the word "lens", as shown by the output of the query below.

```
declare @featureWord char(42)

set @featureWord = 'lens'

select display_term,
       special_term,
       source_term
from   sys.dm_fts_parser('FORMSOF( FREETEXT,"'+@featureWord+'")', 1033,
       null, 0)
```



	display_term	special_term	source_term
1	lens's	Exact Match	lens
2	lensed	Exact Match	lens
3	lenses	Exact Match	lens
4	lenses'	Exact Match	lens
5	lensing	Exact Match	lens
6	lens	Exact Match	lens

Figure 3. Forms of "lens"

Integrating the Full-Text Search functions into the Sentience procedures enables the code, when searching for features, to easily look for not just the feature word itself, but also any form of the feature word that still meets the conditions to be a feature.

For example, if the procedure is looking through a review for opinions related to the “Battery” feature, using the Full-Text Search functions allow the code to understand that comments on the product’s “batteries”, or the “battery’s life”, are also related to the “Battery” feature. The Full-Text Search functionality is integrated into several procedures and functions in Sentience to expand the breadth of the system’s linguistic understanding.

We also integrated WordNet into the process of identifying feature instances. WordNet provides additional abilities for expanding the effectiveness of the code by allowing the code to search for relationships among words. The main relation among words is synonymy, but also has relations coded for hyponymy (type-of relation) and meronymy (part-of relation) [14]. Using WordNet as part of a bootstrapping process in opinion mining was introduced in [8].

Functions and procedures in Sentience incorporate queries against WordNet to find synonymy among words. In the code to identify product features, this expands the words that can be identified for each feature. When finding words in the review that are related to the feature words entered by the user, the system can look for the feature word itself, as well as all forms of the word, and any related words.

When writing the initial code, we first implemented code that would look through WordNet to find both synonyms and hyponyms of the feature words. However, quick reviews of the results made it clear that including hyponymy in the algorithm returned far more false positives in the results than correctly identifying features. Thus, we limited the use of WordNet to identifying the synonymy of related words. We did not, in the research for this thesis, implement any functionality with SentiWordNet or other similar lexicon-based methods for dealing with context dependency. Rather, we implemented rules for context dependency used in [8], as outlined later in this chapter.

Algorithm 1 below provides the general algorithm used in Sentience for identifying instances of product features within reviews.

Algorithm 1. Identify product feature instances

```
features = features listed as input by the user;
for each feature f in features do
    for each review r do
        words = words in review r that are nouns;
        for each word w in words do
            if w is a synonym of f, or a form of f, or a form of a synonym of f do
                feature(w) = f
            endif
        endfor
    endfor
endfor
```

3.3. Determining Semantic Orientation

After the list of features per review has been identified, the next step is to determine the semantic orientation of the opinion held by the reviewer towards the feature. Sentience does this by first finding the descriptor words near each feature instance word per sentence. The system then determines the semantic orientation of each descriptor, and applies this orientation to the related feature.

In the Sentience system, the code can identify either adjectives or opinion verbs as feature descriptors. An example of an adjective as a feature descriptor is:

"The battery life on this phone is *great*."

The following sentence provides an example of an opinion verb used to describe the semantic orientation of the opinion towards the product feature.

"I *love* the picture quality of this camera."

Algorithm 2 shows the steps used in Sentience to build the list of descriptors for each feature instance in a review.

Algorithm 2. Identify feature descriptors

```
for each sentence $s_i$ that contains a set of features do
    $features$ = feature instances contained in $s_i$;

    for each feature $f_j$ in $features$ do
        $words$ = words contained in $s_i$ that are adjectives or opinion verbs;
        for each word $w$ in $words$ do
            if not exists feature $f_o$ between $w$ and $f_j$ then
                $w$ = descriptor($f_j$);
            endif
        endfor
    endfor
endfor
```

3.3.1. A Lexicon-Based Approach to Finding Descriptor Orientation

The base of the sentiment analysis is finding the orientation of feature descriptors, or opinion words, using a lexicon-based approach. Rather than building a comprehensive lexicon of opinion words and their orientation, we selected a limited Orientation Lexicon of opinion words. We then, within the function to find word orientation, used the WordNet synonymy relation of the word to see if the base form of the word, or any synonym of the word, is contained in the Orientation Lexicon.

Function 1: wordOrientation(*word_in*)

```
if word_in is in WordNet then
    word = word_in;
else
    word = FORMOF(word_in) that is in WordNet;
endif

negativelexicon = words and synonyms of words contained in orientationLexicon
    where orientation = -1;
positivelexicon = words and synonyms of words contained in orientationLexicon
    where orientation = 1;

if word is in negativelexicon then
    orientation = -1;
else if word is in positivelexicon then
    orientation = 1;
else orientation  = 0;
endif
```

The lexicon of opinion words used for the wordOrientation() function was constructed by finding words through three methods. First, the authors wrote down as many positive and negative opinion words we could think of. Second, we found online sources with lists of positive [29] and negative [30] opinion words. Additionally, words taken from reviews in our experimental data set were added to the lexicon as they were identified, ensuring that opinion words with a consistent semantic orientation from the reviews were included in our lexicon.

Further research is needed in developing a more comprehensive opinion lexicon that is more widely and generally applicable to more product reviews. Prior research has been done in building effective lexicons [5]. For example, Liu and Hu [31] have provided a lexicon of over 6,800 opinion words that they have compiled over many years of research. Because the lexicon approach to semantic orientation was not a focus of this thesis, but used simply to build a basis of orientation upon which to build with the additional rules, the lexicon we have built works well for its purpose. It has also shown to be fairly effective in determining the orientation of opinion words that have a general semantic orientation.

3.3.2. Linguistic Rules for Context Dependency

After determining the general orientation of the descriptors that are opinion words, we then use the context of the words for additional information on its orientation. These rules, each identified in [8], are implemented in Sentience.

The purpose of these linguistic rules is to identify the semantic orientation of feature descriptors when the orientation cannot be determined just by looking at the general orientation of the word, using the wordOrientation() function. The orientation of many words is dependent on the context in which they are used. Thus, these rules use the context in which the words are used to determine their orientation.

Negation Rule: The negation rule is applied by looking for negation words, such as “no”, “not”, and “never”, used prior to feature descriptors. The rules also consider pattern-based negation such as “stop verbing”, “quit verbing”, and “cease to verb”. If negation words are found prior to the descriptor, then the following rules are applied [8]:

Negation of Negative Orientation → Positive Orientation (e.g., “not bad”)

Negation of Positive Orientation → Negative Orientation (e.g., “not so great”)

Negation of Neutral Orientation → Negative Orientation (e.g. “does not flash”)

Too Rule: The Too Rule is simply that if any descriptor is preceded by the word “too”, then the orientation becomes negative [8]. For example, the orientation of “large” is dependent on the feature it is describing, and cannot be determined on its own. But "too large" can be generally understood to have a negative orientation, independent of the feature it is describing.

But Rule: Sentences that contain the words "but", or the synonyms "however", "with the exception of", "except that", and "except for" provide an additional rule for determining semantic orientation. This rule is based on the linguistic pattern that the word "but" will typically change the semantic orientation of the statement. For example, a person would not typically say "The picture quality is amazing; the flash is low-quality." When a sentence contains more than one opinion with opposing semantic orientations, there is typically a "but" word to indicate the change in orientation: "The picture quality is amazing, but the flash is low-quality." The "but clause" is the opinion phrase beginning with the "but" word. In the previous example, the "but clause" is "but the flash is low-quality." [8]

The rule can also be applied across adjoining sentences. For example:

"The picture quality is amazing. However, I don't think it is worth the high price."

Algorithm 3 shows the logic used in the application of the But Rule in determining semantic orientation.

This rule allows the program to determine the orientation of feature descriptors, where it would otherwise be unknown, by looking at the orientation of other features in the same sentence, or adjoining sentences, where the orientation is known. For example, the word "short" does not, on its own, provide an indication of semantic orientation. Its orientation is dependent on the context in which it is used. When it is used in a sentence such as "The kit lens is incredible, but the battery life is short", then the But Rule can be used to determine its orientation. Because “incredible” is known as having a positive semantic orientation, and it is on the other side of the word “but” in the sentence, then the program can infer that the word “short”, in this context, has the opposite orientation than that of the word “incredible”.

Algorithm 3. But Rule

```
if descriptor word $w_d$ appears in a “but” clause then
    for each unmarked opinion word $ow$ in the “but” clause of sentence $s_i$ do
        if exists another opinion word $ow_i$ in $s_i$ with wordOrientation($ow_i$) $\neq 0$ then
            $orientation$ = wordOrientation($ow_i$);
        end if
    endfor
    if $orientation \neq 0$ then
        return $orientation$;
    else $orientation$ = orientation of the clause before “but”
        if $orientation \neq 0$ then
            return (-1) * $orientation$
        else return 0
        endif
    endif
endif
```

Algorithm 3 shows that, to get the orientation of a word for which we do not know the orientation by previous rules, we first look to other opinion words that are in the same “but” clause with the word. If there is another opinion word or words, we derive the orientation of words of unknown orientation from the orientations already known within the clause. If we are not able to derive the orientation from within the “but” clause, then we look for opinion words before the “but” clause, and use the inverse of their orientation to derive the orientation of the opinion words in the “but” clause. [8]

3.3.3. Conjunction Rules

We have also implemented in Sentience three conjunction rules proposed in [8] that aim to resolve additional challenges with context dependency. These rules are referred to in [8] as a “holistic” approach because they use global information from all reviews, rather than just local information. The rules use contextual information from not only the current review being analyzed, but also other reviews for the same product that have also been analyzed.

Intra-Sentence Conjunction Rule: This rule is based on the use of conjunctions within a sentence to determine the orientation of feature descriptors within the sentence. For example, the following sentence contains a conjunction that joins opinions on two features:

“The lens is spectacular, and its price is very low.”

In this example, the word “low” does not have a general semantic orientation. The Intra-Sentence Conjunction Rule says that the semantic orientation of all opinions expressed in a sentence will be the same direction, unless the direction is changed by a “but” word. It is much more natural to make the statement in the example provided than it is to say the statement below, as our tendency is to keep the orientation the same within a sentence. [8]:

“The lens is spectacular, and the price is very high.”

From the example in the previous paragraph, we can discover that “low” has a positive orientation when used in relation to the feature word “price”. Once this orientation has been determined in one review, the orientation relationship between the two words, or synset, can be applied to understand the orientation of the word in other reviews where the feature descriptor “low” is used to describe the feature “price”. [8]

Pseudo Intra-Sentence Conjunction Rule: It is possible that a sentence may indicate the orientation of a feature descriptor without the explicit use of a conjunction.

This is referred to as the Intra-Sentence Conjunction Rule. For example, consider this sentence:

“The price is low, which is great.”

Although no conjunction is explicitly used in the sentence, the overall orientation of the sentence is positive, due to the use of the word “great”. This positive orientation can be used to infer a positive orientation for the feature descriptor “low” in relation to the feature “price”. As with the Intra-Sentence Conjunction Rule, this can then be applied in other sentences and reviews where the same feature descriptor is used to describe the feature. [8]

Inter-Sentence Conjunction Rule: This rule is an extension of the Intra-Sentence Conjunction Rule to neighboring sentences. It is based on the tendency to follow the same orientation from one sentence to the next. For example one might say:

“The lens is spectacular. The price is very low.”

This would be more natural than to change the orientation between sentences:

“The lens is spectacular. The price is very high.”

If there is a change in orientation between two sentences it is more natural to use a “but” word. For example:

“The lens is spectacular. However, the price is very high.”

The Inter-Sentence Conjunction Rule is applied if the orientation cannot be determined by using the previous two conjunction rules, or the other linguistic rules used for determining context dependency. [8]

Algorithm 4 below provides a view of how the conjunction rules work with the But Rule, as implemented in Sentience, to determine the orientation of feature descriptors where there is context-dependent orientation.

Algorithm 4. Applying conjunction rules

```
for each unmarked feature descriptor $fd_0$ in sentence $s$ do
    if there exists in sentence $s$ another non-neutral feature descriptor $fd_1$ then

        if there is a "but" word between $fd_0$ and $fd_1$ then
            $orientation$ = ($fd_1$).orientation * (-1);
        else $orientation$ = ($fd_1$).orientation;
        endif

    else if sentence ($s$ – 1) exists and has a non-neutral feature descriptor $fd_2$ then
        if the first word of $s$ is a "but" word then
            $orientation$ = ($fd_2$).orientation * (-1);
        else $orientation$ = ($fd_2$).orientation;
        endif

    else if sentence ($s$ + 1) exists and has a non-neutral feature descriptor $fd_3$ then
        if the first word of ($s$ + 1) is a "but" word then
            $orientation$ = ($fd_3$).orientation * (-1);
        else $orientation$ = ($fd_3$).orientation;
        endif

    else $orientation$ = 0;

    endif
endfor
```

3.3.4. Opinion Aggregation

After applying linguistic rules to find the orientation of context-dependent words, the orientation of the feature descriptors must then be applied to the feature to determine the orientation of the opinion holder's view of the feature within the context of the sentence. If there are multiple feature descriptors used to describe the feature in the sentence, then overall orientation for the feature in the sentence is derived based on all feature descriptors for the feature in the sentence.

One of the challenges addressed in [8] is the potential for multiple, conflicting opinion words in the same sentence. To more accurately determine the orientation for each feature instance in a sentence, the authors devised a method for computing the orientation of each feature instance in a sentence, which includes weighting the orientation of each descriptor based on the distance between the feature descriptor and the feature instance with which it is associated. To compute the orientation score for each feature instance, each feature descriptor with a positive orientation is assigned a value of +1. Each feature descriptor with a negative orientation is assigned a value of -1. The following, then, is the score function used to determine the orientation of the feature within the sentence, as written in [8]:

$$score(f) = \sum_{w:w\in s \wedge w\in V} \frac{w.SO}{dis(w,f)} \tag{3.1}$$

where:
w is an opinion word
V is the set of all opinion words
s is the sentence that contains the feature *f*
$dis(w_i, f)$ is the distance between feature *f* and opinion word *w*
w.SO is the semantic orientation of the word *w*

Equation 3.1 gives more weight to feature descriptors that are closer to the feature instance. The authors in [8] found this to be an effective method for working with multiple, and potentially conflicting, opinion words within the same sentence. Opinion words that are farther away from the feature word are less likely to modify the feature. However, there is potential that it will be related. Thus, based on their findings, this method of weighting the opinion words based on distance deals with the challenge well. [8]

As noted in Chapter 2, Mukherjee and Bhattacharyya [21] determined this naïve method of dealing with multiple features and orientations in a sentence to be less effective than providing additional context rules for determining relationships between descriptors and feature instances. However, for the purposes of this thesis, we assume that the rule, as applied in [8], is accurate enough to use as a baseline for our research. We did no additional testing to determine which of the two methods provides the better results.

Algorithm 5 provides an overview of the process to determine the semantic orientation of a feature instance, providing the logic used to apply the linguistic rules discussed in this section of the thesis.

Algorithm 5. Calculating feature orientation score

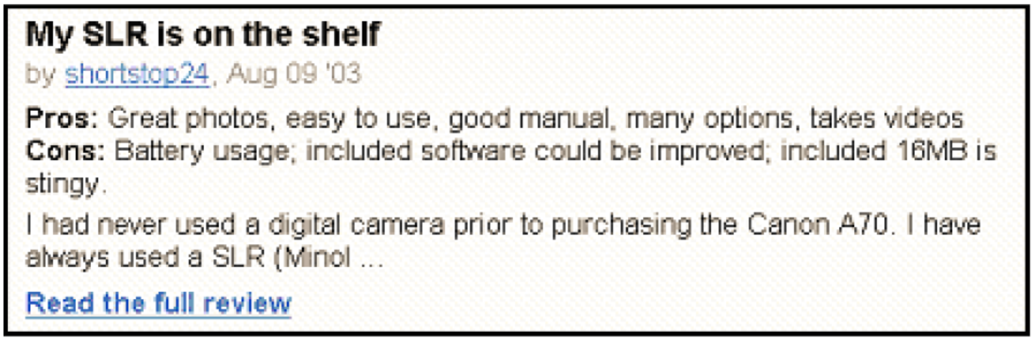
```
for each sentence $s$ that contains a set of features do
    $features$ = feature instances contained in $s$;

    for each feature instance $f$ in $features$ do
        $orientation$ = 0;
        $descriptors$ = identified descriptors of $f$;
        for each descriptor $d_f$ of $f$ do
            $wo$ = wordOrientation($d_f$);
            $wo$ = applyTooRule($wo$);
            $wo$ = applyNegationRule($wo$);

            if $wo$ = 0 then
                $wo$ = applyConjunctionRules($wo$);
            endif

            if $wo \neq 0$ then
                $orientation = orientation + \frac{wo}{dis(d_f, f)}$;
            endif
        endfor
    endfor
endfor
```

3.3.5. Opinion Headings

After using feature descriptors to determine the semantic orientation of each feature per sentence, we next look to additional linguistic rules for each feature to find the orientation of features which have not yet been determined.

One of the contributions of this thesis was the addition of an algorithm to attempt to recognize headings used in the text of a product review. A popular method of writing reviews is to group features by orientation, with a heading for each group providing an indication of the semantic orientation of the group of features. We refer to these as *opinion headings*. For example, the reviewer may have a heading of “Pros” with a section describing the features of the product for which there is a positive opinion. Following the “Pros” section would then be a “Cons” heading, followed by a section describing the features of the product for which there is a negative opinion.

Ganapathibhotla and Liu touched on this topic, discussing that reviewers may use Pros and Cons, but the rules were limited to a format in which the features are in a single, comma-delimited sentence. Figure 4 is an example provided in this research showing the format to which the rules apply. [22]



My SLR is on the shelf
by shortstop24, Aug 09 '03

Pros: Great photos, easy to use, good manual, many options, takes videos
Cons: Battery usage; included software could be improved; included 16MB is stingy.

I had never used a digital camera prior to purchasing the Canon A70. I have always used a SLR (Minol ...

Figure 4. Example review of pros and cons [22]

We see two challenges with basing rules for Pros and Cons on this format. The first is that reviewers listing Pros and Cons will not always put the list of features into a comma-delimited list in a single sentence. There are several potential formats, including a numbered list, bulleted list, a single paragraph, or multiple paragraphs in a section in which each feature is discussed in detail. Each of these potential sections would have a section heading indicating the semantic orientation of the features in the section.

The second challenge is that the section heading for Pros and Cons may not use the words “pros” and “cons”. The headings may use other words with opposing semantic orientation, such as “The Good” and “The Bad”, or “Positives” and “Negatives”. Or there may even be more than two headings, such as “The Good”, “The Bad”, and “The Terrible”. Thus, rules for finding and using headings to determine semantic orientation must be able to:

1. Identify opinion headings.
2. Determine the orientation of the opinion headings.
3. Find the list of features in the section associated with each opinion heading.
4. Apply the orientation of the of the section heading to each feature.

The first step is identifying section headings h_i that may exist in each review r. In looking through a large number of reviews, we found four characteristics that appear to be generally applicable to opinion headings in reviews:

1. The opinion heading is followed by at least one carriage return, separating the opinion heading into its own paragraph p_i.
2. The paragraph is shorter in length than a typical paragraph.
3. The paragraph has a semantic orientation, based on a lexical opinion word.
4. There exists in the review r at least one other short paragraph p_j containing an antonym of the lexical opinion word that appears in paragraph p_i.

Algorithm 6 shows the logic used in the Sentience system for identifying opinion headings, using the characteristics identified above.

Algorithm 6. Identify opinion headings

```
for each paragraph $p_i$ in review $r$ with length <= 42 do
    $words$ = words contained $p_i$;

    for each word $w$ in $words$ do
        $orientation$ = wordOrientation($w$);

        if $orientation \neq 0$ then

            if exists paragraph $p_j$ with length <= 42
                and exists word $w_j$ in paragraph $p_j$ where isAntonym($w$, $w_j$) = 1
                ($w$).$opinionheadingword$ = 1;
            else ($w$). $opinionheadingword$ = 0;
            endif
        else ($w$). $opinionheadingword$ = 0;
        endif
    endfor
endfor
```

The opinion heading rules are then applied after the opinions per feature are calculated based on the descriptors associated with the feature. If, after applying the prior linguistic rules to determine the orientation of the descriptors, and applying these to the feature orientation, the orientation is still unknown, we then apply the opinion heading rules.

Algorithm 7. Apply opinion heading rules to features

```
for each review $r$ that contains opinion headings $h_i$ do
    $features$ = feature instances contained in $r$ with undetermined orientation;

    for each feature instance $f_j$ in $features$ do
        find nearest opinion heading $h_i$ prior to $f_{j;}$
        $orientation$ = wordOrientation($h_i$);
    endfor

    if $orientation$ is null then
        $orientation$ = 0;
    endif
endfor
```

While most of the rules for semantic orientation in the Sentience program are applied to the descriptors, the opinion heading rules are applied to the feature instances. We do not assume that because the feature instance is put under a heading with a specific orientation that all descriptor words under the heading will be of the same orientation. We do assume that the overall semantic orientation applied to the feature instance itself will align with the orientation of the heading.

3.3.6. Informal Forms of Words

An area of research stemming from the application of spell checking in sentiment analysis is that of the use of informal forms of words. Many words have informal forms that are in common usage. For example, “mic” is commonly used to mean “microphone”, and “pic” can be used to mean “picture”. Some informal forms of words, such as “limo” for “limousine” or “tux” for “tuxedo” have become common enough that they are included in the dictionary. But this may not always be the case, and the informal form of a word may not necessarily exist in the dictionary. Without rules to look for informal forms of a word, the system is not able to understand the meaning of the form of the word.

To handle informal forms that do not exist in the dictionary, we implemented in Sentience rules for finding these informal forms. The implemented rules find words in reviews with the following characteristics:

1. The word, including any form of the word, is not in the dictionary.
2. All characters (after dropping the “s” or “es” if the word is plural) of the word with length *n* are the first *n* characters of a feature word.

This is shown in Algorithm 8.

Algorithm 8. Finding informal forms of feature words

```
for each sentence s in review r do
    words = nouns in s that are not found in WordNet

    for each word w do

        if w is plural then
            w = w with the "s" or "es" dropped;
        endif

        if exists a feature word fw in the user feature list that starts with w then
            (w).featureID = (fw).featureID;
        endif

    endfor
endfor
```

3.3.7. Spell Checking

A third contribution of this research in sentiment analysis is in the area of spelling. The goal of this section of the research is to determine whether or not correcting all spelling mistakes in each review will improve the program's ability to find features within the review, and make determinations of opinion orientation.

The Sentience program does not do automated spell checking. There are many challenges to creating an effective system that provides automated spell checking. The system can, fairly easily, determine if a word is not spelled correctly by comparing each word in a review, and all forms of the word, to the WordNet dictionary. However, accurately correcting the spelling of misspelled words as part of an automated sentiment analysis program is more challenging. Current programs, such as Microsoft Word, contain rules that consider word context that can, in many cases, automatically correct a misspelled word. However, there are still scenarios in which there may be several words in the dictionary that are close to the misspelled word, and human intervention is required to determine the correct word to be used in the text.

Tackling these challenges in creating an automated spell-checking system is outside the scope of this research. We aim only to determine to what degree spelling correction affects the accuracy of Sentience.

To perform this analysis, we input into the system the original version of each product review. We also input into the system for each review a version of the review that has had spelling checked and corrected. We are then able to run the sentiment analysis processes against both versions of the reviews, and compare the results to determine whether or not, and to what degree, correcting the spelling in product reviews improves the sentiment analysis performed by the system. This process is described in more detail in Chapter 4.

3.4. Output

The output of the Sentience system is a summary of the semantic orientation per feature for all reviews of the product entered into the system. The overall score for the semantic orientation for each feature is a decimal between -1 and 1 that is an average of the semantic orientation determined by the system.

For our output, we use two algorithms, returning two semantic orientation scores per feature. The first is an average based on each feature instance. This means that we calculate the semantic orientation score per feature by calculating the average of all instances of the feature across all reviews. This is demonstrated in Algorithm 9.

Algorithm 9. Summarizing Sentience results by feature instance

```
features = list of features input by the user to be analyzed;
for each feature f in features do

    featureorientation = average orientation of all instances of feature f;

endfor
```

The second calculation is based on the average of the semantic orientation, when first summarizing the orientation by review. The calculated orientation assigned to each review is determined by taking, for each feature, the average of the orientation calculated for each instance of the feature in the review. For example, consider a review that mentions the feature "shutter speed" four times. In three of the instances the orientation is determined to be positive (+1). In the fourth instance the orientation is determined to be negative (-1). Sentience will summarize the results by averaging the orientations of each instance of the feature, calculating an average orientation of 0.5 for the review.

The feature orientation per review is then set to an integer value, based on the average orientation. If the value is greater than 0, it is determined to be positive, and the feature's semantic orientation for the review is set to +1. If the average orientation is less than 0, it is considered negative, and the feature's semantic orientation for the review is set to -1. If it is 0, then the orientation remains neutral.

After all features are summarized per review, the overall orientation per feature is averaged across all reviews, resulting in a decimal value between -1 and 1. This process is demonstrated in Algorithm 10.

Algorithm 10. Summarizing Sentience results by review

```
features = list of features input by the user to be analyzed;
for each feature f in features do
    featureorientation = 0.0;
    orientationtotal = 0;
    instancecount = 0;

    for each review r that contains feature f do

        orientation = average orientation of all instances of feature f in r;

        if orientation > 0 then
            orientation = 1
        elseif orientation < 0 then
            orientation = -1
        else orientation = 0
        endif

        orientationtotal = orientationtotal + orientation
        instancecount = instancecount + 1

    endfor

    $featureorientation = \frac{orientationtotal}{instancecount}$

endfor
```

The focus of the first calculation, by feature instance, is getting a view of how the system results are affected by differences in individual feature instance scores. The second calculation, by review, is more focused on how the system would typically be used by a company trying to get an understanding of customer sentiment. When first averaging the scores by review, we get an understanding of how each individual review, or opinion holder, feels about each feature. If a reviewer mentions a feature many more times than another reviewer, the second calculation does not give that reviewer any more weight in the overall semantic orientation score.

4. EXPERIMENT AND RESULTS

This section provides the empirical evaluation against the Sentience system to assess the system's accuracy in finding product features within reviews, and determining the semantic orientation of the features within the review. As the goal of the system is to understand and summarize human sentiment, the accuracy of the system must be measured against how well it agrees with how a human would assess the orientation.

4.1. Experimental Process

The overall process for this experiment involves four steps:

1. Obtaining product reviews.
2. Manually reading and annotating the product reviews, determining the features discussed in each review, and the semantic orientation of the opinion held by the reviewer for each product feature. The results of this process are referred to as *human judgment* in this thesis.
3. Using Sentience to analyze the same product reviews, determining the features discussed in each review, and the semantic orientation of the opinion held by the reviewer for each product feature.
4. Compare the features and orientation results found by the Sentience system against human judgment. The accuracy of the Sentience system is determined by how close its results are to that of human judgment.

The first step is obtaining the product reviews. For this experiment, the input into the Sentience system is a set of product reviews for two different products on Amazon.com (http://www.amazon.com). We used a three-step process to get the data into the system for analysis.

The first step for each review was to do part-of-speech tagging on the review using the NLProcessor [25]. The NLProcessor tags each word and word group with a part of speech. The part-of-speech information is used in several Sentience procedures, as noted in Chapter 3 of this thesis.

The second step is to check the spelling in the review. We used the spellchecking functionality in Microsoft Word to assist in this process, but each review was read manually to verify and correct the spelling. The NLProcessor system was then used to do part-of-speech tagging against the version of the review with corrected spelling.

Third, the review was read to identify features in each sentence, and determine the semantic orientation of each feature within the context of the sentence, as understood by the authors. This resulted in a list of feature instances per sentence for each review, with the orientation of each feature instance identified as positive, negative, or neutral.

Table 2 provides a summary of the review data used for the reviews.

Table 2. Characteristics of the Product Review Data

	dSLR Camera	Blu-Ray Player	Total
No. of reviews	50	24	74
Total words	31,105	6,801	37,906
Avg. words per review	622	283	512
Total spelling corrections	109	46	155
Spelling corrections/1000 words	3.5	6.76	4.09
Total feature instances	671	140	811
Avg. feature instances per review	13.42	5.83	10.96

4.2. Measuring System Performance

As in [8], we use the standard evaluation measures of precision (*p*), recall (*r*), and F-score (*F*) to measure the performance of the system.

$$F = \frac{2pr}{p+r} \tag{4.1}$$

Precision is the fraction of the orientation results retrieved that are relevant, while recall is the fraction of the relevant system-calculated orientation instances that are retrieved [32]. The F-score is the harmonic mean of precision and recall [33].

For our system assessment, we ran the system against the reviews in five combinations of rule application, or scenarios. For each of the scenarios, which are outlined in more detail below, we find the precision, recall, and F-score of three types. The first calculation type is to find the relevance of the algorithm used to identify features in the product reviews. With this calculation, we want to determine how well the system can identify instances of features, when compared to the feature instances identified by human judgment. This calculation looks only at the existence of feature instances, and does not include the semantic orientation for feature instances. For this calculation, the following are how precision (*pFI*) and recall (*rFI*) are determined:

$$pFI = \frac{HJF \cap SF}{SF} \tag{4.2}$$

$$rFI = \frac{HJF \cap SF}{HJF} \tag{4.3}$$

where:
pFI = precision of feature instance identification by Sentience
rFI = recall of feature instance identification by Sentience
HJF = set of feature instances identified by human judgment
SF = set of feature instances identified by Sentience

For example, consider the following data sets of human judgment feature instances with orientation, and Sentience-identified feature instances with identified orientation, as given in Table 3 and Table 4:

Table 3. Human Judgment Features (*HJF*) with Orientation

Review ID	Sentence ID	Feature	Orientation
1	4	Price	+1
1	8	Lens	-1
2	7	Lens	+1
2	14	Sensor	+1
3	5	Price	-1
5	16	Battery	0

Table 4. Sentience-identified Features (*SF*) with Orientation

Review ID	Sentence ID	Feature	Orientation
1	4	Price	0
1	8	Lens	-1
2	7	Lens	-1
2	14	Sensor	+1
4	12	Flash	+1

In this example, of the five features identified by the system, four were also contained in the list of features identified by manually reading the reviews. So the precision (*pFI*) would be 4/5, or .8. The recall (*rFI*) would be 4/6, or .67.

The second F-score calculation assesses how well the system can identify the orientation of feature instances. For this calculation, we are using the list of identified features from the previous calculation, and adding the semantic orientation to the equation.

$$pO = \frac{HJO \cap SO}{SO} \tag{4.4}$$

$$rO = \frac{HJO \cap SO}{HJO} \tag{4.5}$$

where:
pO = precision of feature instance orientation identified by Sentience
rO = recall of feature instance orientation identified by Sentience
HJO = set of feature instance orientation identified by human judgment
SO = set of feature instance orientation identified by Sentience

Consider the features and orientations identified in Table 3 and Table 4. Of the five feature instances identified by the system, four of them were also identified by human judgment. Of the four correct feature instances in the system-identified set, two of the feature instances have the same orientation as the associated, human judgment orientation. Thus, the *pO* is 2/5, or .4. The *rO* is 2/6, or .33.

The third F-score calculation assesses how relevant the Sentience results are when summarizing all feature orientations per feature, by review. The goal is to see if the system can, for an individual review, determine how the opinion holder for the review feels about each feature after summarizing the results of semantic orientation per feature within each review, as outlined in Algorithm 10.

$$pRO = \frac{HJRO \cap SRO}{SRO} \tag{4.6}$$

$$rRO = \frac{HJRO \cap SRO}{HJRO} \tag{4.7}$$

where:

pRO= precision of the identification of feature orientation by review performed by Sentience

rRO= recall of the identification of feature orientation by review performed by Sentience

HJRO= set of feature semantic orientation identified by human judgment, summarized by review

SRO = set of feature semantic orientation identified by Sentience, summarized by review

For example, consider the following data sets of human judgment feature instances with their orientation, and Sentience-identified feature instances with their system-calculated orientation, as given in Table 5 and Table 6.

Table 5. Human Judgment Features with Orientation (*HJRO*)

Review ID	Sentence ID	Feature	Orientation
1	1	Price	+1
1	2	Lens	-1
1	3	Price	+1
2	1	Lens	+1
2	2	Lens	+1
2	3	Sensor	-1
2	4	Sensor	0

Table 6. Sentience-identified Features with Orientation (*SRO*)

Review ID	Sentence ID	Feature	Orientation
1	1	Price	0
1	2	Lens	-1
1	3	Price	-1
2	1	Lens	+1
2	2	Lens	+1
2	3	Sensor	-1
2	4	Flash	0

The following shows the results of summarizing the feature orientation for each review (Table 7):

Table 7. Summary Feature Orientation by Review

Review ID	Feature	Human Judgment Orientation	Sentience Orientation
1	Price	+1	-1
1	Lens	-1	-1
2	Lens	+1	+1
2	Sensor	-1	-1
2	Flash	NULL	0

With this example, the *rRO* is 3/5, or .6. Of the five feature orientations identified by the system, three of the five matched the feature orientations per review identified by human judgment. The *rRO* is then 3/4, or .75. Because the system, in this example, found a feature instance that was not found by human judgment, the precision will be lower than the recall.

As discussed earlier, each of the three F-score calculations was used in five scenarios to test the effectiveness of the individual rules that were contributions to the research in this thesis (Opinion Heading Rule, Informal Forms of Words Rule, and Spell Checking). The following are the five scenarios run:

1. The Sentience system was run, applying all three of the new rules to identify features and determine semantic orientation.
2. Of the three new rules, only the Heading Rule was applied.
3. Of the three new rules, only Spell Checking was applied.
4. Of the three new rules, only the Informal Forms of Words Rule was applied.
5. None of the new rules were applied.

For each of these scenarios, all of the rules that were adapted from previous research, as identified in Chapter 3 of this thesis, and implemented in Sentience, were applied for the scenario. These rules adapted from previous research are used as a base to assess whether or not adding the new rules will improve the accuracy of the system.

4.3. Experiment Results

Table 8, Table 9, and Table 10 show the results of the precision, recall, and F-score calculations for each of the five scenarios, by product.

Table 8. Feature Instance-based F-Score Calculations by Product

	dSLR Camera			Blu-Ray Player			
Scenario	Precision	Recall	F-score	Precision	Recall	F-score	p-value
1. All New Rules	0.780	0.797	0.789	0.847	0.871	0.859	0.02663
2. Heading Rules	0.773	0.768	0.770	0.846	0.864	0.855	N/A
3. Spell Check Applied	0.776	0.781	0.779	0.847	0.871	0.859	0.00960
4. Informal Word Form Rule	0.778	0.787	0.782	0.846	0.864	0.855	0.06555
5. No New Rules Applied	0.773	0.768	0.770	0.846	0.864	0.855	N/A
Average	**0.776**	**0.780**	**0.778**	**0.847**	**0.867**	**0.857**	**N/A**

Table 9. Feature Instance Orientation-based F-Score Calculations by Product

	dSLR Camera			Blu-Ray Player			
Scenario	Precision	Recall	F-score	Precision	Recall	F-score	p-value
1. All New Rules	0.382	0.390	0.386	0.375	0.386	0.380	.00436
2. Heading Rules	0.375	0.373	0.374	0.371	0.379	0.375	.03780
3. Spell Check Applied	0.382	0.385	0.383	0.375	0.386	0.380	.00088
4. Informal Word Form Rule	0.373	0.377	0.375	0.371	0.379	0.375	.16110
5. No New Rules Applied	0.374	0.371	0.372	0.371	0.379	0.375	N/A
Average	**0.377**	**0.379**	**0.378**	**0.372**	**0.381**	**0.377**	**N/A**

Table 10. F-Score Calculations by Product for Summarized Orientation by Review

	dSLR Camera			Blu-Ray Player			
Scenario	Precision	Recall	F-score	Precision	Recall	F-score	p-value
1. All New Rules	0.621	0.617	0.619	0.552	0.529	0.540	.00896
2. Heading Rules	0.619	0.585	0.601	0.545	0.514	0.529	.03785
3. Spell Check Applied	0.608	0.585	0.596	0.552	0.529	0.540	.00358
4. Informal Word Form Rule	0.611	0.601	0.606	0.545	0.514	0.529	.07389
5. No New Rules Applied	0.607	0.574	0.590	0.545	0.514	0.529	N/A
Average	**0.613**	**0.592**	**0.602**	**0.548**	**0.520**	**0.534**	**N/A**

For each of the three calculation types, we used a *t*-test to determine whether or not the application of each rule provides a change in the precision, recall, and F-score that has statistical significance. Table 8, Table 9, and Table 10 each contain the p-value resulting from the test, when comparing the calculated precision, recall, and F-score values of each scenario to the scenario in which no new rules were applied.

In addition to calculating the F-score for each test scenario, we also looked at the results of the system output for each scenario of each product. The system output provides four data sets for each feature that are calculated semantic orientation scores. These are scores calculated by the system on a scale between -1 and 1 showing the overall degree of sentiment across all reviews for each product feature. These are the scores discussed in Algorithm 9 and Algorithm 10 in Section 3.3.7.

The first two scores produced as output are based on feature instances as outlined in Algorithm 9. Figure 5 below provides an example of the system output when displaying the scores based on feature instance. The figure shows the calculated orientation scores for the Blu-Ray Player product, when run with no new rules applied (Scenario 5), and calculating the orientation scores by feature instance.

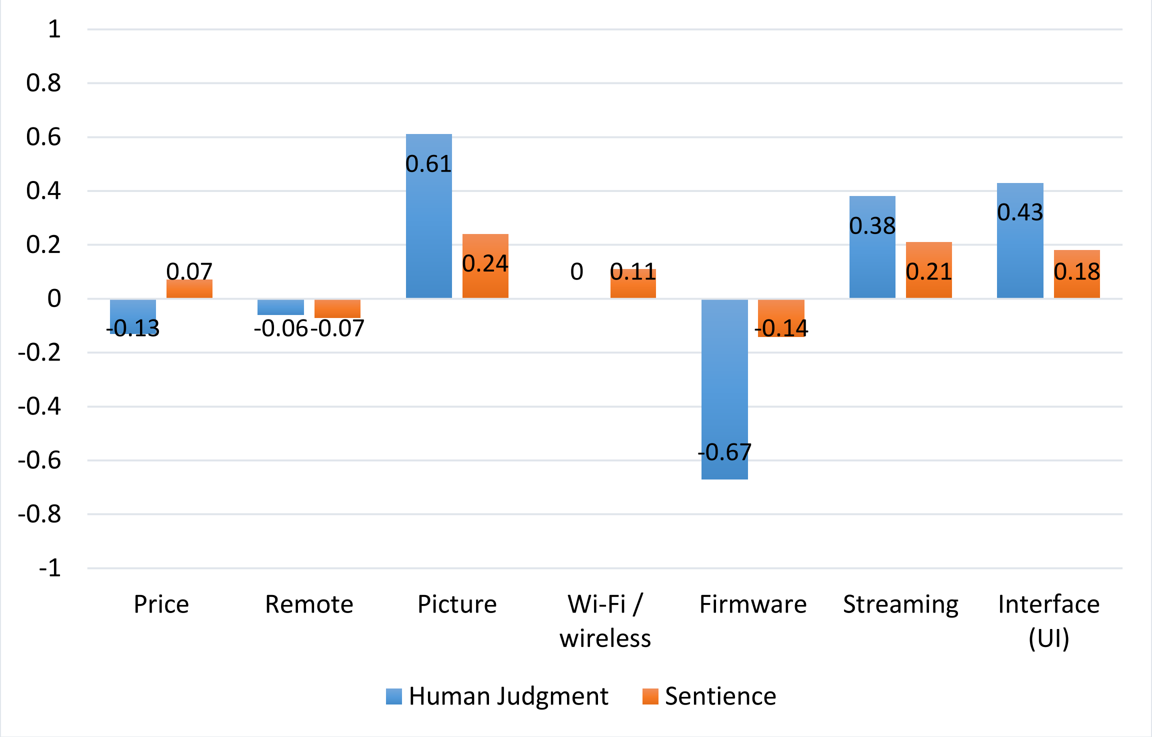


Figure 5. Feature semantic orientation scores, human judgment versus Sentience

The first score is calculated for average human judgment semantic orientation based on feature instance. The second is the average feature instance semantic orientation score, calculated by Sentience. These two numbers displayed together show the comparison in calculated overall orientation scores per feature for the product between what the system calculates, and the calculated orientation scores based on human judgment.

The next set of semantic orientation scores, calculations three and four, are based on the scores when summarizing the orientation by review prior to calculating the final orientation per feature. This calculation is shown in Algorithm 10. Using Algorithm 10, the third calculation, similar to the first calculation, is based on the orientation by human judgment. The fourth calculation, similar to the second, is calculated based on the Sentience-determined orientation.

After calculating the orientation scores by feature instance and review for both the manual and Sentience calculations, we looked at the differences between the scores for each feature to see how the average difference between the Sentience system and manual calculations were affected by the different scenarios of rule application. Figure 6 below shows the average difference between the human judgment semantic orientation score, and the Sentience-calculated semantic orientation score for each of the five scenarios, for both calculation types for both products. The two calculation types are for feature instance (FI) and by review (R).

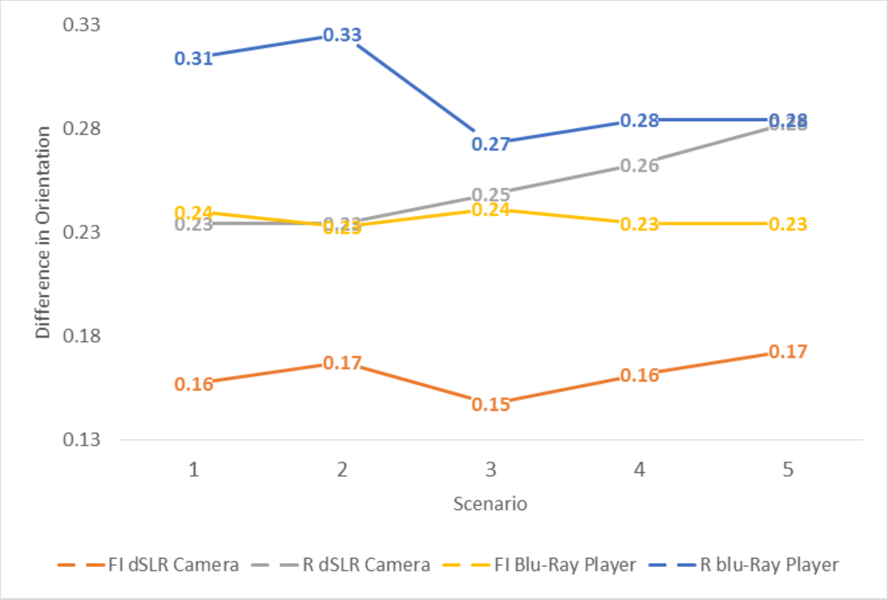


Figure 6. Average difference in semantic orientation score

4.4. Discussion

Overall, the experiments above show that the added rules provide slight increases in the effectiveness of the sentiment analysis system, with some situational dependencies.

4.4.1. Opinion Headings

The first proposed rule is to search for opinion headings within each review that express an opinion related to features that exist in the section below the heading. For this rule, the goal was to improve the accuracy with which the system can identify the semantic orientation of each feature instance within the product reviews. This rule is applied after the list of feature instances is created by the system, so it has no effect on the effectiveness with which the system is able to identify the feature instances.

The results of the experiment show that the rule provides a slight increase in the effectiveness of the Sentience system in identifying the semantic orientation of feature instances, depending on the product. With the dSLR Camera, adding the opinion heading rule provided an increase in both precision (from .374 to .375) and recall (from .371 to .373) over the test scenario with no new rules applied.

However, when running the system against the reviews for the Blu-Ray Player, there was no change in either precision or recall when compared to the test scenario with no new rules applied. In reviewing the results, there was only a single review for the Blu-Ray Player product in the data that was identified as having a heading. In that case, the orientation of the features contained in the section had already been identified using previous rules, leading to no change in the F-score for the application of the rule.

Although the rule provided no change in F-score for the Blu-Ray Player product, and only a slight change for the dSLR Camera product, the p-score calculated for the *t*-test does show that there is a statistically significant increase in the score after applying the rule.

The effectiveness of this rule appears to have a correlation with the complexity of the product being reviewed. The dSLR Camera product seems to have more features and attributes on which reviewers have comments than the Blu-Ray Player product, leading to a lengthier, on average, review. The average dSLR Camera product review has over twice as many words as the average review for the Blu-Ray Player product. From this observation, it appears that longer product reviews will have a greater tendency to have opinion headings separating the review in the sections, oriented by opinion.

Across the two products, the average length of reviews that have opinion headings is significantly larger than reviews that do not have opinion headings, as shown in Table 11 below.

Table 11. Comparison of Review Length, Reviews with Headings versus No Headings

		No. of Reviews	Avg. Word Count
dSLR Camera	Reviews with headings	7	1280
	Reviews without headings	43	515
Blu-Ray Player	Reviews with headings	1	439
	Reviews without headings	23	277

In future work regarding this rule, a larger data set across a larger number of products will be required to assess the rule’s general applicability and accuracy. However, for the purposes of this thesis, we find that applying the opinion heading rule does provide a small measure of improvement in the analysis of semantic orientation where opinion headings are used within the reviews.

The application of the opinion heading rule had a greater impact on precision and recall when summarizing feature sentiment by review, as shown in Table 10. This indicates that the increases in precision and recall per feature instance were typically enough to affect the overall feature orientation per review where the rule affects feature instances.

4.4.2. Informal Forms of Words

We expected that the rule for finding the informal forms of words would affect the accuracy of both the process to find feature instances, as well as the process to determine the semantic orientation of each feature instance. We anticipated that the greater impact would be related to the functionality of identifying feature instances, as the informal words used appeared to be related to features rather than feature descriptors.

This rule had a positive impact in both precision and recall on the dSLR Camera product when calculating the F-score based on feature instances, based on feature instance orientation, and based on reviews. Of the three new rules proposed, this rule had the greatest impact in precision and recall for the dSLR Camera product when calculating the F-score based on feature instance and reviews.

However, with the Blu-Ray product, applying this rule had no effect on either precision or recall for any of the three F-score calculations. When reviewing the results, we found that the applied rules did not find any informal words. In manually reading the reviews, we did not find any words in the reviews that we felt should have been affected by the rule. Figure 6 also demonstrates this. In the figure, for the Blu-Ray Player product there is no change in the difference between the Sentience-calculated score and the score calculated based on human judgment orientation, for either the feature instance or review summary calculations, between the scenario with no new rules applied (Scenario 5) and the scenario with only the Informal Forms of Words rule applied (Scenario 4).

With these results, it appears that the rule can be effective, depending on the product. Some products will have specialized terminology and informal forms of words that are common to the product, while other products will not. We feel that this shows some rules have situational effectiveness. As with the heading rule, more testing is needed in future research to prove the general applicability of this rule across a wide array of products. The p-value calculated for the *t*-test is greater than .05, showing that the application of the rule does not provide a statistically significant increase in the scores. We feel that there may be potential for this rule to be effective in specific scenarios, but our testing does not show that the rule has general effectiveness across products.

4.4.3. Spell Checking

The goal of spell checking the reviews was to see whether or not doing so would improve the accuracy of the system. We anticipated, as with the Informal Forms of Words Rule, that applying spell checking would affect both the process of identifying feature instances, as well as determining the semantic orientation of each feature instance, as the change to the spelling is applied prior to beginning both processes in Sentience.

With this test, we found that correcting the spelling in the reviews had a positive, and statistically significant impact, improving both precision and recall in identifying feature instances, and determining the semantic orientation of feature instances, for both products. Of the three new rules tested, correcting the spelling proved to be the rule that demonstrated the greatest impact on the F-score for each product when calculating the orientation based on feature instance.

We anticipated that there would be some improvement in both the precision and recall, but we did not anticipate that the impact of spelling correction would be greater than that of the other rules. When initially reviewing the data for the corrections made in each review, the number of spelling mistakes did not appear to be very significant; and the number of spelling mistakes directly related to product features and descriptors of product features appeared to be even less significant.

However, upon review of the results, we found that the impact of spelling correction came primarily due to improvements in the natural language processing functionality of the NLProcessor program, as well as the parsing functionality in Sentience. In the instances where there are spelling mistakes in the text, the NLProcessor is unable to understand the part of speech, and is unable to correctly apply the tags. This incorrect tagging affects not only the misspelled word, but also tagging of the rest of the sentence in which the word is found. By correcting a single spelling error, the parsing functionality can work correctly, and multiple features and/or descriptors can be affected.

5. CONCLUSIONS

5.1. Summary

In this thesis, we provided an overview of sentiment analysis, including an introduction to research done in the field of study up to this point in Chapter 2. We found, in reviewing the prior research in sentiment analysis, that there is still room for improvement in the field, and areas of study for additional work. We created, for this thesis, a software system that builds on the sentiment analysis rules of previous work, and adds proposed rules for improving the effectiveness of the system. The developed system is outlined in Chapter 3. In Chapter 4, we explained our process for assessing the effectiveness of the system, and provided the experimental results.

5.2. Conclusion

In this thesis, we proposed three areas of improvement for the effectiveness of sentiment analysis systems. Two of the areas included algorithms for linguistic conventions, allowing the computer system to understand sentiment expressed in natural language more closely to how it is understood by human judgment. The research also included an evaluation of spelling to assess any increase in effectiveness based on improved spelling.

Across all three proposed areas of research for this thesis, we saw situational effectiveness. Each of the rules proposed provided some measure of improvement in at least some scenarios, showing that the rules can provide improvement in sentiment analysis systems.

However, the increase in effectiveness was limited. There is significant room for improvement in the system. We have identified several areas requiring future work and research, and there are likely many more. Computerized understanding of language is an interesting challenge, and there is a lot of room for future growth and research.

5.3. Future Work

Although a lot of research has been done in the area of natural language processing, and feature-based sentiment analysis, there is still a lot of improvement that can be done to bring a computer system's understanding of the language closer to that of a human's. There are several areas of research that can be pursued to improve the effectiveness and usefulness of the Sentience system, and sentiment analysis systems in general.

5.3.1. Feature Input

The first area in which there can be improvement in Sentience is in automatically determining the features that are discussed in each review without the need for the user to input the features for which the system should search. In the initial planning for the Sentience system for this thesis, the design decision was made based on the concept of convenience for the end user to be able to identify the product features about which they are most concerned. However, in reviewing the results of the implemented system, we feel that there could be a significant improvement in the system's ability to determine the semantic orientation of feature instances if all feature instances are identified for all potential features, rather than just a subset of features.

The reason for the improvement would come through the rules that determine descriptor orientation for context-dependent words. The implemented rules can determine the orientation of a descriptor with unknown orientation by looking at surrounding descriptors in the same or neighboring sentences. However, descriptors are only identified as descriptors if they are related to a feature. Limiting the list of features, and thus feature instances, also limits the potential feature descriptors that can be used to determine context dependency.

To enhance the system in this area, Sentience should have automated functionality for searching through the reviews, and identifying all features discussed by the reviewer, and grouping the features by synonyms.

We recognize that having the system automatically identify features in each review, rather than having the user define the feature for which the system should search, will create additional challenges with the Informal Forms of Words rule defined in this thesis. Because our current system implementation requires the user to define the features for which the system will search, the defined features will be related to the product. The Informal Forms of Words rule looks for words that are similar to the features defined by the user. Without the list defined by the user, the rule does not have a frame of reference with which to determine whether or not the informal form is relevant within the context of the review. For example, the rule, as currently applied, could identify the word “spec” as being an informal form of “specification”, “specialist”, or several other words, if they were to be entered as a feature by the user, returning a false positive as a result of the rule.

Thus, creating rules for the system to automatically determine the features will require additional rules in searching for informal forms of words to ensure that the informal forms found are related to the context of the review. For example, a possible additional rule may be to only consider informal forms of the word valid if the full word is found in the review, or another review of the same product. For example, the word “mic” would only be considered a valid informal form of the word “microphone” if the full word “microphone” is found in the same review, or another review of the same product. This, or other rules would need to be tested, to ensure effective rules are developed for accurately finding and grouping features for each product and review.

There are other challenges to accurately identifying and grouping features in documents, as discussed by Zhai, et al. For example, it is not enough to simply group synonyms together as related features as words that are not necessarily synonyms can still describe the same feature. Also, as with other domains, there can be context-dependent synonymy [23]. Improvements to Sentience’s ability to identify and classify feature can be made in future work by building on this research.

5.3.2. Spelling

A second potential area for future research is in implementing spell-checking algorithms to automate as much spelling correction as possible in sentiment analysis systems. One of the contributions of this research was to show that correcting spelling mistakes in documents prior to performing sentiment analysis does provide improvements in the system’s accuracy. A next step is to determine methods that are able to automatically correct spelling mistakes as a step in the sentiment analysis process.

5.3.3. Pronouns

Another area of improvement for future work is in identifying feature instances based on pronouns, such as “it”, “these”, or “they”. Sentience currently has no rules to understand or identify the feature words associated with pronouns in the sentence. The rules for identifying feature words by pronoun have the potential to be fairly complex. Pronouns can be used to substitute feature words within a single sentence, such as:

“The lens is great for daytime pictures, but *it* struggles in low light.”

In this case, the rules should be able to identify that “it” is referencing the “lens” feature, and that the sentence is expressing two opposing opinions related to the same feature. Pronouns may also be used across multiple sentences. A feature word may be used at the beginning of a paragraph, and pronouns could then be used in the place of the feature word throughout the rest of the paragraph.

There is also the potential for multiple feature words to be used within the sentence or sentences in which there is a pronoun used. It is possible that the pronoun may be used to substitute one, or multiple feature words. Consider these two examples:

“The audio quality is okay, but the picture quality and video quality are much better.
They are both beautiful.”

“The camera has video and audio recording. They are both impressively sharp.”

In the first example, the pronoun refers to two of the three features in the prior sentence. In the second example it refers to both. Additional scenarios would also need to be considered in researching the rules for how a computer would understand the use of pronouns. The system must understand the language rules and linguistic conventions associated with the rules.

5.3.4. General Functionality Terms

One of the areas in which Sentience is unable to understand the features being discussed is in terms that are used generally to describe the primary functionality of the product. For example, a review might state:

“Even in low light you get awesome results.”

Human judgment would understand that the reviewer is referencing the primary functionality of the camera: taking pictures. The word “results” is most likely in reference to the picture quality provided by the product in low light.

There are other general words such as “performance” and “outcome” that are used to describe the primary product functions. A potential area of future work is to research rules for understanding features discussed by general product performance terms. Creating accurate rules that are generally applicable across products could be a significant challenge, as terminology used can be different for reviews of different products. However, it is an area where research can be done to determine if rules can be written that can be generally applied across products and review topics.

5.3.5. Implicit Features

The system should be able to identify implicit features discussed by the user. This is a problem that was identified in [8] and discussed earlier in this thesis. But it does not appear that an adequate solution has yet been developed for this problem. The challenge is creating general rules that can understand a feature based only on descriptors of the feature. For example, a reviewer stating that a product is “expensive” is expressing an opinion on the feature “cost” of the product, or stating that the product is “huge” is expressing an opinion on the feature “size”, without explicitly identifying the feature.

The solution to the problem of implicit features may be to develop a new lexicon, similar to the work with WordNet and other extensions to WordNet, that identifies word relationships based on features and descriptors. There may be other solutions or potential linguistic rules to solve the problem, but additional research is needed in this area.

"Price" was an implicit feature, in particular, with which Sentience struggled to accurately identify the orientation of feature instances. It is a feature for which it is more common to make references using both implicit features and implicit opinions. For example, rather than explicitly stating that the reviewer likes the price of the product, the reviewer might state:

"You get all of this for less than $700."

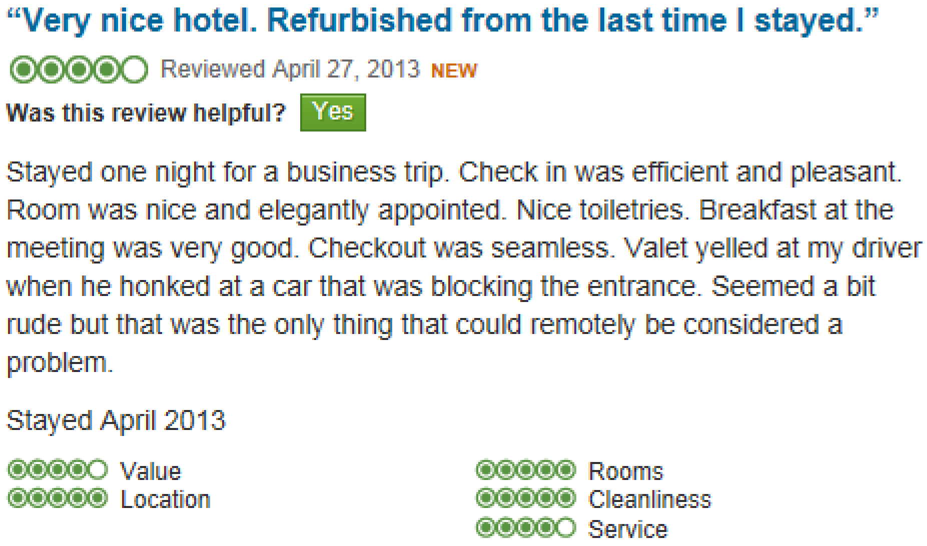
Future work can be done around the concepts of implicit features and opinions to allow the system to better understand the features and orientations discussed.

5.3.6. Feature Importance Factor

The rules and discussion in this thesis are related to improving a system's ability to identify feature instances in product reviews, and determine the semantic orientation of the opinion held by the reviewer for the feature. A potential area for future work in feature-based sentiment analysis is in determining the importance of those opinions in the reviewer's overall view of the product.

With feature-based sentiment mining, the system will produce a list of features discussed by the reviewer, as well as the semantic orientation, or how the reviewer feels, about each feature. But there is potential to have the system also provide a score on how the opinion of each feature affected the overall rating of the product.

Vu, Li, and Beliakov [34] propose a method for determining which product features in a product review have the greatest weight in determining the customer's overall satisfaction with the product. In their research, the authors use hotel reviews mined from Trip Advisor (http://www.tripadvisor.com). The reviews, as shown in Figure 7 below, have pre-defined product features that the customer is able to rate, as well as provide an overall rating for the hotel.



“Very nice hotel. Refurbished from the last time I stayed.”

●●●●○ Reviewed April 27, 2013 NEW

Was this review helpful? Yes

Stayed one night for a business trip. Check in was efficient and pleasant. Room was nice and elegantly appointed. Nice toiletries. Breakfast at the meeting was very good. Checkout was seamless. Valet yelled at my driver when he honked at a car that was blocking the entrance. Seemed a bit rude but that was the only thing that could remotely be considered a problem.

Stayed April 2013

●●●●○ Value
●●●●● Location
●●●●● Rooms
●●●●● Cleanliness
●●●●○ Service

Figure 7. Hotel review from Trip Advisor [35]

Using the ratings provided, the authors built a model based on the Choquet integral to determine which features have the greatest impact on the overall product sentiment. In the example provided in Figure 7, the product features “value” and “service” were each rated 4 out of 5, while “location”, “rooms”, and “cleanliness” were each rated 5 out of 5. And the overall rating for the hotel was 4 out of 5. Thus, we can assume for this customer that “value” and “service” are the most important product features. The model proposed by the authors can then aggregate this information for large numbers of reviews to provide a summary of the most important features, and how customers feel about those features. [34]

This principle can be applied to systems that do feature-based opinion mining. The model developed by Vu, Li, and Beliakov assumes that the reviews will have product features predefined, and that each reviewer will rate the product features as given by the site. However, many e-commerce sites do not have pre-defined product categories and features for each product on the site. Thus, the feature set and sentiment orientation for each product feature need to be determined by parsing the text-based reviews using the methods and linguistic rules discussed in this thesis. Adding this model to the sentiment analysis system can open up more business use cases, allowing businesses to see not only how customers feel about product features, but also how important that feature is to the customer. This allows the business to understand which features to focus on for enhancements, as well as which features to highlight in marketing campaigns.

5.3.7. Degrees of Sentiment

The model using the Choquet integral developed in [34] is based on the principles of fuzzy logic, which provides for assigning degrees of membership to the elements in a set [36]. A crisp set assigns membership in the group providing two options for each element: either the element is a member of the group (1), or is not a member of the group (0). Fuzzy logic allows for degrees of membership in the group between 0 and 1.

In the sentiment analysis system, this can be applied by allowing for degrees of sentiment in the analysis. Current sentiment analysis models view feature-based sentiment orientation with a crisp point of view, assigning the sentiment orientation based on positive (+1), negative (-1), or neutral (0). There is no allowance for degrees of membership. In the business use cases in [34], degrees of membership are a necessary aspect of the model because they allow the model to understand the degree of importance of each product feature. Rather than a 0 or 1, each feature in [34] is rated on a scale.

To improve the effectiveness of the predictive model, degrees of sentiment should be measured and assigned for each feature instance. For example, a review using the terms "most incredible ever" to describe a feature would be assigned a higher sentiment rating in relation to the feature than a review stating that the feature is “pretty good”. With current sentiment analysis models, both of these examples would both be given the same rating of +1, as they are both considered positive reviews of the feature.

Assigning a degree of feature sentiment to each review provides a significant challenge. There is still a lot of work needed to accurately assign each feature instance a sentiment score on the current crisp scale. Adding degrees of sentiment into the equation makes this more complicated. However, there is research that can be done in this area.

Thelwall, et al. [11], provided some insight into this topic with their research into sentiment strength detection. In their work, they propose linguistic rules for determining the strength of the sentiment expressed when analyzing comments on MySpace (http://www.myspace.com). As discussed in Chapter 2, the core of the algorithm used in their work is a lexicon of words created, including a word strength score. Each word in the lexicon is labeled as either positive or negative, with an additional score from 1 to 5 indicating the sentiment strength. A few additional rules were also used to refine the results determined by applying the lexicon. [11]

Another potential area of research in determining degrees of sentiment is in the use of comparative and superlative forms of words. Ganapathibhotla and Liu [22] provided some research into the use of comparative and superlative forms of words in sentiment analysis. The context of their research involved mining opinions from sentences that provide comparisons between products. For example, “the sensor in this model is on par with the APS-C model EOS-7D.” Their work provides a model for the computer system to understand the product and feature preferences based on the comparisons to other products and features.

Research can also be done to apply the work of comparative and superlative forms of words to understanding degrees of sentiment. A model could be built to assign a degree of sentiment based on the form of the word used. For example, the word “good” may receive a sentiment score in the lexicon of .5. Using comparative and superlative forms of the words could then escalate the orientation score to, for example, .7 for “better”, and 1.0 for “best”.

6. BIBLIOGRAPHY

[1] L. Perner, "Consumer Research Methods," University of Southern California, 2010. [Online]. Available: http://www.consumerpsychologist.com/cb_Research_Methods.html. [Accessed 8 February 2014].

[2] B. Pang, L. Lee and S. Vaithyanathan, "Thumbs up? Sentiment Classification using Machine Learning Techniques," *Proceedings of EMNLP 2002,* pp. 79-86, 2002.

[3] B. Liu, "Sentiment Analysis and Subjectivity," in *Handbook of Natural Language Processing*, Boca Raton, CRC Press, 2010.

[4] P. D. Turney, "Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews," in *40th Annual Meeting of the Association for Computational Linguistics*, Philadelphia, 2002.

[5] M. Hu and B. Liu, "Mining Opinion Features in Customer Reviews," in *Nineteenth National Conference on Artificial Intelligence*, San Jose, 2004.

[6] M. Ogneva, "How Companies Can Use Sentiment Analysis to Improve Their Business," 19 April 2010. [Online]. Available: http://mashable.com/2010/04/19/sentiment-analysis/. [Accessed 8 February 2014].

[7] C. Tan, L. Lee, J. Tang, L. Jiang, M. Zhou and P. Li, "User-Level Sentiment Analysis Incorporating Social Networks," in *Proceedings of KDD 2011*, San Diego, 2011.

[8] X. Ding, B. Liu and P. S. Yu, "A Holistic lexicon-Based Approach to Opinion Mining," in *WSDM '08*, Palo Alto, 2008.

[9] B. Liu, Sentiment Analysis and Opinion Mining, Morgan & Claypool Publishers, 2012.

[10] A. Denis, S. Cruz-Lara and N. Bellalem, "General Purpose Textual Sentiment Analysis and Emotion Detection Tools," *arXiv,* vol. 1309, no. 2853v1, 2013.

[11] M. Thelwall, K. Buckley, G. Paltoglou and D. Cai, "Sentiment Strength Detection in Short Information Text," *Journal of the American Society for Information Science and Technology,* vol. 61, pp. 2554-2558, 2010.

[12] "Paul Ekman," Wikipedia, 12 March 2004. [Online]. Available: http://en.wikipedia.org/wiki/Paul_Ekman. [Accessed 19 February 2014].

[13] A. Meyers, "Analytics for Twitter 2013," Microsoft, 11 December 2013. [Online]. Available: http://social.technet.microsoft.com/wiki/contents/articles/15665.analytics-for-twitter-2013.aspx. [Accessed 15 February 2014].

[14] The Trustees of Princeton University, "WordNet - A Lexical Database for English," Princeton University, 7 November 2013. [Online]. Available: http://wordnet.princeton.edu/. [Accessed 11 February 2014].

[15] M. Guerini, L. Gatti and M. Turchi, "Sentiment Analysis: How to Derive Prior Polarities from SentiWordNet," in *EMNLP*, Seattle, 2013.

[16] "SentiWordNet," 2010. [Online]. Available: http://sentiwordnet.isti.cnr.it/. [Accessed 19 February 2014].

[17] A. Esuli and F. Sebastiani, "SentiWordNet: A Publicly Available Lexical Resource for Opinion Mining," in *LREC-06, the 5th Conference on Language Resources and Evaluation*, Genova, 2006.

[18] K. Paramesha and K. Ravishankar, "Optimization of Cross Domain Sentiment Analysis Using SentiWordNet," *International Journal in Foundations of Computer Science & Technology,* vol. 3, no. 5, 2013.

[19] C. Strapparava and A. Valitutti, "WordNet-Affect: an Affective Extension of WordNet," in *Proceedings of LREC*, Lisbon, 2004.

[20] E. Cambria, C. Havasi and A. Hussain, "SenticNet 2: A Semantic and Affective Resource for Opinion Mining and Sentiment Analysis," in *Proceedings of the Twenty-Fifth International Florida Artificial Intelligence Research Society Conference*, Marco Island, 2012.

[21] S. Mukherjee and P. Bhattacharyya, "Feature Specific Sentiment Analysis for Product Reviews," *Computational Linguistics and Intelligent Text Processing,* vol. 7181/2012, pp. 475-487, 2012.

[22] M. Ganapathibhotla and B. Liu, "Mining Opinions in Comparative Sentences," 2008.

[23] Z. Zhai, B. Liu, H. Xu and J. Peifa, "Clustering Product Features for Opinion Mining," in *Fourth ACM International Conference on Web Search and Data Mining*, Hong Kong, 2011.

[24] N. Archak, A. Ghose and P. Ipeirotis, "Deriving the Pricing Power of Product Features by Mining Consumer Reviews," *Management Science,* vol. 57, no. 8, pp. 1485-1509, 2011.

[25] Infogistics, "NLProcessor - Text Analysis Tookit," 2001. [Online]. Available: http://www.infogistics.com/textanalysis.html. [Accessed 2 February 2014].

[26] Microsoft, "Full-Text Search (SQL Server)," Microsoft, 2014. [Online]. Available: http://technet.microsoft.com/en-us/library/ms142571.aspx. [Accessed 10 February 2014].

[27] Microsoft, "sys.dm_fts_parser," 2014. [Online]. Available: http://technet.microsoft.com/en-us/library/cc280463.aspx. [Accessed 10 February 2014].

[28] Microsoft, "FREETEXT (Transact-SQL)," 2014. [Online]. Available: http://technet.microsoft.com/en-us/library/ms176078(v=sql.105).aspx. [Accessed 10 February 2014].

[29] "An Energizing List of Positive Words," The Benefits of Positive Thinking, 2013. [Online]. Available: http://www.the-benefits-of-positive-thinking.com/list-of-positive-words.html. [Accessed 10 September 2013].

[30] S. Hein, "Negative Feeling Words," Emotional Intelligence, 2013. [Online]. Available: http://eqi.org/fw_neg.htm. [Accessed 10 September 2013].

[31] B. Liu and M. Hu, "Opinion Mining, Sentiment Analysis, and Opinion Spam Detection," University of Illinois at Chicago, 15 May 2004. [Online]. Available:

http://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html. [Accessed 12 February 2014].

[32] "Precision and recall," Wikipedia, 11 January 2014. [Online]. Available: http://en.wikipedia.org/wiki/Precision_and_recall. [Accessed 7 January 2014].

[33] "F1 score," Wikipedia, 18 January 2014. [Online]. Available: http://en.wikipedia.org/wiki/F1_score. [Accessed 7 January 2014].

[34] H. Q. Vu, G. Li and G. Beliakov, "A Fuzzy Decision Support Method for Customer Preferences Analysis based on Choquet integral," in *WCCI 2012 World Congress on Computational Intelligence*, Brisbane, 2012.

[35] "Trip Advisor," 27 April 2013. [Online]. Available: http://www.tripadvisor.com. [Accessed 8 December 2013].

[36] P. Hajek, "Fuzzy Logic," The Stanford Encyclopedia of Philosophy (Fall 2010 Edition), 2010. [Online]. Available: http://plato.stanford.edu/archives/fall2010/entries/logic-fuzzy. [Accessed 15 February 2014].