June 2020

Sentiment Analysis in Peer Review

Zachariah J. Beasley

University of South Florida

Follow this and additional works at: https://scholarcommons.usf.edu/etd

Part of the Computer Sciences Commons

Scholar Commons Citation


This Dissertation is brought to you for free and open access by the Graduate School at Scholar Commons. It has been accepted for inclusion in Graduate Theses and Dissertations by an authorized administrator of Scholar Commons. For more information, please contact scholarcommons@usf.edu.
Sentiment Analysis in Peer Review

by

Zachariah J. Beasley

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Computer Science and Engineering
Department of Computer Science and Engineering
College of Engineering
University of South Florida

Major Professor: Les Piegl, Ph.D.
Paul Rosen, Ph.D.
Yao Liu, Ph.D.
Susana Lai-Yuen, Ph.D.
Alon Friedman, Ph.D.

Date of Approval:
May 29, 2020

Keywords: Natural Language Processing, Text Mining, Aspect Extraction, Scalable Assessment, Crowdsourcing

Copyright © 2020, Zachariah J. Beasley
Dedication

To the true Author and Perfecter. To my parents, Robert and Elizabeth, who trained me in the way I should go. To my wife, Camille, who inspires me daily with her love and faith. And finally, to my daughter, Esther, who invokes such joy.
Acknowledgments

While at the University of South Florida, I was very fortunate to have studied under several distinguished and supportive faculty members. I would like to express my sincerest thanks to Dr. Les Piegl, Dr. Paul Rosen, Dr. Yao Liu, Dr. Susana Lai-Yuen, and Dr. Alon Friedman for kindly serving on my dissertation committee and offering constructive feedback on my manuscript. I am very grateful to Dr. Friedman and Dr. Rosen for their assistance in applying my work to their data visualization courses and for their mentorship. I would especially like to thank Dr. Piegl, my major professor, for his advice, support, encouragement, and mentorship. I could not have asked for a better advisor.

I am also grateful for my labmates and friends, from whom I learned much: Dr. William Oropallo, Dr. Zhila Nouri, Dr. Junyi Tu, Ghulam Quadri, TJ Kotha, Dr. Sulav Malla, Anwesh Tuladhar, Dr. Cagri Cetin, Dr. Jean-Baptiste Subils, Steven Diaz, and Jorge Adorno. I would like to express my gratitude to Dr. Tu for formatting support and Dr. Subils for sharing his dissertation format template.

I would like to thank my pastors, Bill and Margie Cooper, and all my friends from church, including Kirk and Nancy Allen and Chris and Kristi Baker, for their kindness and support. I am so thankful to my family, Robert, Elizabeth, Isaac, Andrea, Nathanael, Abigail, Elijah, David, Minerva, David Michael, and Kelli, for their love for me and interest in my work. I am grateful to my mother for instilling in me a love of learning through homeschooling and to my father for exemplifying academic excellence at the highest level. This dissertation would certainly not have been possible without my wife, Camille, who encouraged, challenged, supported, and loved me. “She opens her mouth with wisdom, and the teaching of kindness is on her tongue.” Finally, I would like to thank Jesus Christ, my Lord and Savior, for His blessing and providence (Matthew 6:33).
# Table of Contents

List of Tables iv
List of Figures v
Abstract vii

Chapter 1: Introduction 1
  1.1 Motivation 1
  1.2 Application 2
  1.3 Terminology 3
  1.4 Background 5
  1.5 Advantages and Disadvantages 6
  1.6 Contribution 7
  1.7 Organization 8

Chapter 2: Sentiment Analysis 9
  2.1 Core Components 10
    2.1.1 Data 10
    2.1.2 Score 12
    2.1.3 Architecture 13
    2.1.4 Transfer Learning 16
  2.2 Sentiment in Education 18
    2.2.1 Determining Helpfulness by Content 20
    2.2.2 Improving Helpfulness by Matching 21
    2.2.3 Creating Helpfulness by Good Questions 21
    2.2.4 Balancing Review Burden and Fairness 22
  2.3 Review Form 19
    2.3.1 Determining Helpfulness by Content 20
    2.3.2 Improving Helpfulness by Matching 21
    2.3.3 Creating Helpfulness by Good Questions 21
    2.3.4 Balancing Review Burden and Fairness 22

Chapter 3: Theoretical Framework 24
  3.1 State of Scalable Assessment 25
    3.1.1 Assignment Type 25
    3.1.2 Assessment Options 26
  3.2 Educational Principles 27
    3.2.1 Learning Environment 27
    3.2.2 Review Form Questions 29
3.2.3 Scaling
3.2.4 Data

Chapter 4: Aspect Extraction
4.1 Review Form
4.1.1 Review Form Components
4.1.2 Intelligent Data Combing
4.1.3 Interpreting Questions
4.1.4 Results
4.2 Aspect Extractor
4.2.1 Design
4.2.2 Validation
4.2.2.1 External
4.2.2.2 Internal

Chapter 5: Lexicon
5.1 Process
5.2 Polarity
5.2.1 Direction
5.2.2 Weight
5.3 Other Lexicons
5.3.1 Synsets
5.3.2 Word-Value Pair
5.3.3 Formatting and Cleaning
5.4 Experiment
5.4.1 Weighted or Unweighted
5.4.2 Matching Form Score
5.4.3 Lexicon Information
5.4.4 Qualitative Example

Chapter 6: Sentiment Analysis
6.1 Algorithm
6.1.1 Negation
6.1.2 Assigning a Grade
6.1.3 Grading Example
6.1.4 Comparing Grading Schemes
6.1.5 Discussion
6.1.6 Supporting Documentation
6.2 Evaluation
6.3 Sampling
6.4 Run Time
6.4.1 Email Scraping
6.4.2 Subjective and Analytical Graders
List of Tables

Table 4.1  Review Process Iterations 40
Table 4.2  Restaurant Aspect Extractor 44
Table 4.3  Restaurant Aspects not Previously Found 45
Table 5.1  Sample of Lexicon Words 48
Table 5.2  Weighted Versus Unweighted Lexicon 56
Table 5.3  Weighted Lexicon Comparison 57
Table 5.4  Lexicon Information from Graduate Courses 57
Table 5.5  Lexicon Information from Undergraduate Courses 58
Table 6.1  Metric Correlations with Mean and Median 72
Table 6.2  Graduate Review Sampling 78
Table 6.3  Undergraduate Review Sampling 78
Table 7.1  Analytic Evaluation of Peer Comments 95
Table A.1  Figure License Information 127
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Assessment Process</td>
<td>2</td>
</tr>
<tr>
<td>1.2</td>
<td>Google Ngram for “Sentiment Analysis”</td>
<td>6</td>
</tr>
<tr>
<td>2.1</td>
<td>Classification</td>
<td>13</td>
</tr>
<tr>
<td>2.2</td>
<td>Regression</td>
<td>14</td>
</tr>
<tr>
<td>2.3</td>
<td>Basic Neural Network</td>
<td>15</td>
</tr>
<tr>
<td>2.4</td>
<td>Recursive Neural Tensor Network</td>
<td>16</td>
</tr>
<tr>
<td>3.1</td>
<td>Organic Learning</td>
<td>28</td>
</tr>
<tr>
<td>3.2</td>
<td>Learning Environment</td>
<td>29</td>
</tr>
<tr>
<td>3.3</td>
<td>Learning Process</td>
<td>30</td>
</tr>
<tr>
<td>3.4</td>
<td>Knowledge Sampling</td>
<td>31</td>
</tr>
<tr>
<td>3.5</td>
<td>Course Scaling</td>
<td>32</td>
</tr>
<tr>
<td>4.1</td>
<td>Seed-Growing Algorithm for Review Form</td>
<td>36</td>
</tr>
<tr>
<td>4.2</td>
<td>Sliding Window</td>
<td>41</td>
</tr>
<tr>
<td>4.3</td>
<td>Example Aspects</td>
<td>41</td>
</tr>
<tr>
<td>4.4</td>
<td>Aspect Summary for a Single Work</td>
<td>42</td>
</tr>
<tr>
<td>5.1</td>
<td>Word Cloud of Semester Key Words</td>
<td>50</td>
</tr>
<tr>
<td>5.2</td>
<td>Polarity of Tokens</td>
<td>51</td>
</tr>
<tr>
<td>5.3</td>
<td>Qualitative Comparison of Lexicons</td>
<td>60</td>
</tr>
<tr>
<td>6.1</td>
<td>Lexicon Check</td>
<td>63</td>
</tr>
<tr>
<td>6.2</td>
<td>Grading Process</td>
<td>66</td>
</tr>
<tr>
<td>6.3</td>
<td>Sentiment Scoring Method</td>
<td>68</td>
</tr>
<tr>
<td>Figure 6.4</td>
<td>Sentiment Scoring Example</td>
<td>69</td>
</tr>
<tr>
<td>Figure 6.5</td>
<td>GM 2019 Change in Score</td>
<td>72</td>
</tr>
<tr>
<td>Figure 6.6</td>
<td>SE 2019 Change in Score</td>
<td>73</td>
</tr>
<tr>
<td>Figure 6.7</td>
<td>Team Review Summary Example</td>
<td>76</td>
</tr>
<tr>
<td>Figure 6.8</td>
<td>Average Grade Error for Sampling Levels</td>
<td>79</td>
</tr>
<tr>
<td>Figure 6.9</td>
<td>Average Run Time Scaling Reviews</td>
<td>81</td>
</tr>
<tr>
<td>Figure 6.10</td>
<td>Average Run Time Scaling Projects</td>
<td>82</td>
</tr>
<tr>
<td>Figure 7.1</td>
<td>Histograms of Questionnaire Respondent Demographics</td>
<td>89</td>
</tr>
<tr>
<td>Figure 7.2</td>
<td>Survey Results for “How Well” Peer Review Interfaces Worked</td>
<td>91</td>
</tr>
<tr>
<td>Figure 7.3</td>
<td>Dashboard and Qualitative Feedback</td>
<td>92</td>
</tr>
<tr>
<td>Figure 7.4</td>
<td>Histograms of Questionnaire Respondent Rubric Feedback</td>
<td>93</td>
</tr>
<tr>
<td>Figure 7.5</td>
<td>Visualization of Peer Review Sentiment</td>
<td>94</td>
</tr>
<tr>
<td>Figure 7.6</td>
<td>Histograms of Questionnaire Respondent Engagement and Effort</td>
<td>96</td>
</tr>
<tr>
<td>Figure 7.7</td>
<td>Histograms of Questionnaire Respondent Perceived Utility of Feedback</td>
<td>98</td>
</tr>
<tr>
<td>Figure 7.8</td>
<td>Histograms of Questionnaire Respondent Perceived Quality of Feedback</td>
<td>98</td>
</tr>
<tr>
<td>Figure 7.9</td>
<td>Student Improvement Receiving Feedback</td>
<td>99</td>
</tr>
<tr>
<td>Figure 7.10</td>
<td>Student Improvement Giving Feedback</td>
<td>101</td>
</tr>
</tbody>
</table>
Abstract

Sentiment analysis, a widely popular subfield of natural language processing, has recently been used in the classroom to predict student attrition or to determine the mood of students, teacher strengths and weaknesses, or student perception of internship experience. These are all helpful indicators for the enhancement of students’ academic experience but none improve the information gathered from or the reliability of peer review. This is particularly important in large courses with complex assignments (e.g., essays, software projects, and presentations) where scalable grading is requisite. In this dissertation, we apply sentiment analysis not on an assignment itself, but on the meaningful content generated by a learning, peer-reviewing crowd to produce a fine-grained, quantitative score from peer review text alone.

To obtain a reliable score from peer review text, we first supply an educational framework that increases the amount of critical feedback students provide. We then utilize an aspect extractor to aggregate pertinent information from student content and modify our review form in a data-driven, iterative fashion. HeLPS, our domain-specific lexicon, was mined from peer review comments and exhibits high precision on peer review text compared to other publicly-available lexicons. Our sentiment analysis algorithm, SentiSoft, leverages both the lexicon and aspect extractor to provide a fine-grained sentiment score with metrics and supporting documentation from text alone. The combination of sentiment analysis on text and an iteratively-refined review form improves our ability to understand student feedback and ultimately facilitates scalable assessment.
Chapter 1: Introduction

1.1 Motivation

Scalable grading of complex assignments (e.g., websites, essays, designs, and open-ended questions) is notoriously hard in large, online academic environments [1], [2], [3], [4]. For example, UC San Diego Professor Scott Klemmer has taught online courses in Human-Computer Interaction on Coursera that have garnered over 3,600 students, with weekly website design tasks. For an individual or small team to grade such an open-ended assignment is effectively impossible. To solve this scalability problem while maintaining a project-based architecture, we (like Dr. Klemmer and others) employ the widely-used evaluation methodology of peer review. However, peer review can be unreliable due to a low number of reviewers (typically 3-5) and an un-evaluated review form. A scalable, reliable, data-driven, and crowd-sourced assessment solution that provides direct and timely feedback is necessary.

Our work addresses both scalability and reliability in a novel way by organizing our courses to incorporate a reviewing crowd and leveraging sentiment analysis in the peer review process. Rather than apply natural language processing (NLP) techniques to a work itself (e.g., an Automated Essay Scorer like [5]), we apply the techniques to the meaningful content generated by a peer-reviewing crowd (Figure 1.1). Our lexicon-based approach for sentiment analysis of peer review text allows us to 1) collect sentiment to derive a fine-grained grade from peer review text, 2) verify that grade with various metrics for the instructor, and 3) discover aspects important to the reviewers which are candidates for addition to the review form. This data-driven process allows us to gather an increased amount of information from
the peer review process, which can provide the instructor with a better understanding of peer review feedback, increase confidence in assigning a grade, and ultimately move towards removing the instructor/TAs as the grading bottleneck of complex assignments in massively open online courses (MOOCs).

**Figure 1.1: Assessment Process**

1.2 Application

Great care was taken to ensure the generalizability and applicability of our work. We therefore propose a process, not a tool. Thus, although this work was performed at the collegiate level, the principles found are applicable in an educational setting at any level where students are mature enough to present information to one another (peer teaching) and assess their understanding of the information (reviewing). Finally, our process requires a threshold of students to leverage the *wisdom of the crowd* (i.e., the idea that many diverse opinions, when aggregated properly, can outperform an expert judgment, a concept related to the law of large numbers) [6], [7]. In his 2018 letter to shareholders, Amazon CEO Jeff Bezos observes this phenomenon at play in independent third-party seller growth from 3% of Amazon business in 1999 to 58% in 2018 (first-party business grew by 25% over the same period) [8]. In his book, *The Wisdom of Crowds: Why the Many Are Smarter Than the Few*...
and How Collective Wisdom Shapes Business, Economies, Societies and Nations, Surowiecki underscores five principles that a crowd must exhibit to be considered “wise”: Diversity of opinion (students are individual beings and perceive projects differently), Independence (students fill out review forms individually), Decentralization (students are not taught how to fill out the review form or what to say in the open-ended comments), Aggregation (reviews are aggregated by simple and transparent techniques), and Trust (students understand that they will in turn be reviewed and that the instructor can override any grade deemed overly harsh) [6]. Surowiecki does not mention the required number of individuals to constitute a crowd. However, in our context of computer science and engineering courses, which include peer teaching and crowd-sourced peer review [9], we found 15-20 reviewers required for sufficient stability to confidently assess a student’s work (chapter 6), though we typically had 30-40 student reviewers per work.

Aside from the assessment component, peer review has been found to have a number of useful side effects, including stimulating learning in reviewing students [10], [11] and affording students an opportunity to engage with course content by communicating recently-learned material to one another [12]. Thus, our theoretical framework (chapter 3) is of standalone utility, even if our entire process of leveraging NLP techniques in the classroom is not followed exactly.

1.3 Terminology

Throughout this dissertation, the term peer review will be used in lieu of peer assessment, although both are, for all practical purposes, synonymous. Importantly, peer review in our context is neither meant to correspond to an instructor’s ground truth grade nor is it for the purpose of revision and re-submission (as in scholarly/academic/journal reviews). Instead, it is a post-publication review on a final product meant to capture the opinion/intent of the reviewing students. Although all of our research was undertaken at the tertiary educational
level, we use the term *instructor* rather than professor to broaden the accessibility of our work.

Rather than delve into a philosophical discussion on the absolute truth score of a student’s work, we simply compare crowd score and instructor score without mentioning which is correct. Although the instructor determines the final grade, even experts are fallible. Indeed, Zhang lists seven common human rater errors, some of which are intensified with fatigue: severity/leniency, scale shrinkage, inconsistency, halo effect, stereotyping, perception difference, and rater drift [13]. In contrast, a crowd has the potential to average out such errors, and studies have even shown that asking individuals the same question over a period of time — “polling the crowd within” — produces a better average result than asking once [14]. As another example, in the once-popular television show, Who Wants to Be a Millionaire, the contestants’ chosen expert correctly guessed the answer only 65% of the time, in contrast to the “ask the audience” answer, which was right 91% of the time [6]. Thus, it is our belief that *combining* information from the crowd with an instructor’s score will ultimately result in a clearer picture of the grade a student’s work should receive. This is similar to the concept of ensembles in machine learning, which uses a variety of learning algorithms to exploit the advantages of each. Interestingly, in five semesters of use (nine courses, over three hundred student group submissions), the instructor only changed the algorithm’s suggested letter grade once. Again, this does not verify the correctness of the algorithm score, but it does show its agreement with the instructor score, which can be interpreted as one indicator of reliability.

The term *lexicon* will refer to our sentiment analysis dictionaries, since the word is more specific to the domain of NLP. Our lexicons separate sentiment-bearing words into different categories (positive, negative, etc.) as detailed in chapter 5. We also use the term *review form* over rubric. This is to match peer *review* explicitly and to highlight that it is for the students’ use, not the instructor’s. The review form is used as a summative assessment tool
in our context, although an example is provided in chapter 7 where a different framework and a formative review form are used along with our sentiment analysis algorithm to deliver insights into how students benefit from the peer review process. Work will refer to a student group’s submission, where a group is comprised of 2-5 students. In our courses, works include a 1) in-class presentation, 2) academic essay, and 3) open-ended term project.

1.4 Background

Research on sentiment analysis began in the early 2000s following prior research on sentiments and opinions in text and is today a top research area in NLP [15], [16]. Figure 1.2 shows the Google Books Ngram Viewer graph for the term “sentiment analysis”. The graph shows the monotonic growth of references to the term in books published between 1980 and 2008, with the default smoothing of three. Sentiment analysis has a variety of applications, both academic and commercial, in social media (e.g., Twitter), creative works (e.g., Rotten Tomatoes), and products (e.g., Amazon). There are a number of reasons for the proliferation of sentiment analysis research, including the increasing availability of opinionated text on the world wide web and the tendency of businesses, politicians, marketers, and consumers to seek others’ opinions before making a decision [15].

In brief, sentiment analysis (or opinion mining or subjectivity analysis) is a branch of NLP which is used to determine subjective opinion (positive, negative, or neutral) towards a target (physical or theoretical object) in text. Polarity refers to the direction and strength of subjective opinion, and targets are comprised of aspects. The process of aspect extraction collects aspects and their associated sentiment from text to determine to a finer level of detail the writer’s sentiment (in contrast to sentence- or document-level sentiment).

Although sentiment analysis exists in the educational domain, no work has 1) leveraged sentiment analysis in the peer review process to provide or validate grades from text or 2) utilized aspect extraction to develop a review form based on what students have actually
communicated. The second application is important because review forms typically have an open-ended comment section for a free form response (implicitly acknowledging that review forms may not capture *everything* students are trying to communicate). This information can be used to infer whether a review form is missing important questions (hence, aspect extraction). The first application is important because, although an instructor might read the comments if they are short and the course is small, the likelihood of gathering any useful information decreases as the course size grows. At some point information is lost in the overwhelming noise and comments are ignored, rather than summarized.

![Figure 1.2: Google Ngram for “Sentiment Analysis”](image)

1.5 Advantages and Disadvantages

There are a number of key advantages to using both our framework and sentiment analysis in the peer review process. The sentiment score can be a robust check on the score provided by the analytical portion of the review form. This increases confidence that the review form correctly determines a score from the reviewing student’s selections. Further, if students can only repeat terms from the analytical portion of the review form, it increases confidence that the review form correctly captures the entirety of the reviewing student’s thoughts. These two pieces of information are first steps towards empirically evaluating a review form,
although much work remains to be done. Finally, by providing metrics and a summary to the instructor, there is additional justification for the score a work receives. In summary, our approach can provide extra information to 1) improve the peer review process and 2) qualitatively justify a peer review grade (particularly helpful in MOOCs where the instructor may not review every work).

There are also, however, some disadvantages to our process. Building the initial review form (chapter 4) can require creativity and experience, and bootstrapping the lexicons by hand from students’ comments (chapter 5) takes substantial effort and focus (since this must be completed before the aspect extractor can be utilized). Further, since this is an iterative process, revising the review form necessitates some recurring labor and beginning iterations of the review form might be less than perfect. Thus, one must be willing to invest initial effort to experience the later benefits of a robust process.

1.6 Contribution

The main contribution of this work is the description and evaluation of a novel process: the development of a domain-specific lexicon and sentiment scorer to equip an instructor to build a review form with aspect extraction and to grade peer review comments through sentiment analysis. It is not a one size fits all review form, thus we do not provide our review form and encourage others to use it. Neither is it a one size fits all lexicon. Lexicons should closely match their domain — reviewer comments in a creative writing course may not match those of a software testing course (e.g., our software testing lexicon does not penalize the words “error” or “bug” since finding a bug is positive in that context). We hope to provide a starting point for researchers to expand our work, educators to employ our process, and students to benefit from our framework.
1.7 Organization

This dissertation is organized as follows: chapter 2 outlines the related work of sentiment analysis in the education domain and contrasts it to the social media, creative work, and product review domains. Chapter 3 defines the theoretical framework for and organization of our courses, which include crowd-sourced peer review augmented by sentiment analysis. Chapter 4 illustrates the iterative, data-driven process by which we built our review form. Chapter 5 describes our domain-specific, weighted lexicon, the process by which it was built, and the advantages over lexicons available in other domains. Chapter 6 details our sentiment analysis algorithm. Chapter 7 discusses the wider applicability of our process, presenting an example of its use in a course with a different framework and focus. Finally, chapter 8 offers directions for future research, summarizes, and concludes the dissertation.
Chapter 2: Sentiment Analysis

“When dealing with people, remember you are not dealing with creatures of logic, but with creatures of emotion.” – Dale Carnegie

In this chapter, we describe the process of and options for sentiment analysis and provide the related work. Because sentiment analysis is used to predict ratings from text in a large variety of domains, we attempt to cover representative examples from social media, creative works (e.g., movies), and products. In an ideal world, we could simply take the best trained classifier from one of these categories and use it to determine student grades. However, there are a number of complications to this approach, detailed specifically in subsection 2.1.4.

When highlighting others’ approaches, we report on either accuracy, precision, recall, or F1 score according to the evaluation method prioritized in the author’s own work. Accuracy refers to the percentage of examples classified correctly. Precision is the number of true positives over the sum of true positives and false positives (i.e., the ability to prevent false positives). Recall is the number of true positives over the sum of true positives and false negatives (i.e., the ability to find all true positives). The F1 score is the harmonic mean of (giving equal importance to) precision and recall:

\[
F1 \text{ score} = 2 \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}
\]  

Section 2.1 in this chapter describes core components that determine sentiment algorithm architecture from the type of data available, type of score generated from subjective content,

1 Portions of this chapter are from [9]. Permission is included in Appendix A.
and the applicability of transfer learning (taking proven algorithms from one domain and using them in another). Section 2.2 details examples of sentiment analysis in the education domain and contrasts the applications with our own process. Section 2.3 provides an overview of research on the review form, helpfulness of review comments, and review process in general.

2.1 Core Components

When extracting numerical values for sentiment in text, there are a variety of options that stem firstly from the type of data available.

2.1.1 Data

One of the most fundamental determiners of architecture, scoring method, and possibility of transfer learning is whether or not the data is labeled. Labeled data contains a ground truth score associated with each data point by which algorithms can learn to predict and measure closeness to an objective standard. Labels can be assigned either by hand or automatically from features in the text (e.g., smiling or frowning emoticons in a Twitter dataset [17]). Supervised learning (classification and regression) and reinforcement learning (maximizing reward) are two categories of learning on labeled data. If no labels are available, the category of unsupervised learning is appropriate, with clustering as one of the main methods. Clustering groups data by similarities within the structure of data (e.g., grouping reviews into two categories based on a distance metric then perhaps labeling one category ‘positive’ and the other ‘negative’). Finally, there is a fourth category, semi-supervised learning, which is applied when data is largely unlabeled with a few examples of labeled data.

Our data was comprised of review form textual comments, with an associated review form score. This numerical score was aggregated from radio button selections (chapter 4). The review form score was our first candidate for ground truth of the text, however, even if we assumed a perfect review form and process of extracting score (both unlikely in beginning
iterations), the text often did not match the review form sentiment (a phenomenon also observed in the product reviews domain [18]). For example, a student might have selected ‘Overall excellent’ and then proceed to write a highly negative review because the team presentation was boring or teammates read off of the slides. Thus, we found students used the analytic and subjective parts of the review form differently, precluding us from using the review form score as a ground truth. Others have also discovered this discrepancy when training a lexicon-based classifier with a positive, negative, and neutral class on 5.1 million reviews of over 20,000 Amazon products [19]. Although they achieved a fairly high accuracy on manually labeled sentences, their algorithm performed very poorly (an F1 score less than 0.5) when attempting to match the label a customer actually chose.

Secondly, we considered using the instructor score as ground truth. This was problematic as a single ground truth for every individual’s textual response. Many students, learning the material for the first time, presumably commented on fewer aspects of the work than the instructor, who had a better sense of the big picture. It was also possible that a student found something the instructor missed. Ultimately, we could not expect each student’s review to conform to the instructor’s.

Thus our peer review text was unlabeled and we did not train our algorithm using any ground truth per student review. This helped to prevent a biased system (mitigating the seven common human rater errors that apply equally to instructors and students [13]) and allowed us to resolve differences between the wisdom of the crowd and the expert in context as they arose. We did, however, compare the aggregate of student’s review scores in hindsight to the instructor score and found over five semesters of use (nine courses, more than three hundred student group submissions) that the instructor only changed the algorithm’s suggested letter grade once. This showed incredible agreement, and hinted at the reliability of the crowd’s score without using labeled training data.
2.1.2 Score

A score for subjective content can be determined at the document-, sentence-, phrase-, or aspect-level. An aspect is an entity about which some subjective opinion is expressed. Algorithms must often aggregate more specific sentiments (e.g., on aspects or phrases) to produce a document-level score, which can be less reliable since it is less focused [20]. There are three scoring methods depending on the data and the desired output: clustering, classification, and regression.

Clustering in unsupervised or semi-supervised learning is the process of grouping similar data points and later providing a score. Clustering was inadequate for our problem domain because we needed the difference between reviews to leverage the wisdom of the crowd (others note the importance of observing every instance of positive or negative sentiment in a review [21]). Furthermore, each review was unique and we could not determine a priori how many clusters to form.

Classification (Figure 2.1) in supervised learning can be categorical (e.g., ‘happy’, ‘sad’, ‘helpful’), ordinal (e.g., ‘low’, ‘medium’, ‘high’), or integer-valued (e.g., 1, 2, or 3 stars). Classification is perhaps the most used method for scoring text because of the wealth of datasets that are labeled as finite integer values (e.g., 2-, 3-, or 5-level) [15], [22]. Although increasing the number of classes allows better division of sentiment, it also increases the problem difficulty — the number of classes are inversely related to the classifier’s accuracy. Some researchers in the creative work domain even propose learning a classifier per reviewer ([23] and [24], with a neural network approach), but we are focused on a single method of aggregation for revolving student reviewers. All these forms of classification, however, were also unsuitable for our problem domain, since the score a student work receives must be real-valued. Especially when a particular grade was weighted towards a final course grade, we could not simply give the student an ‘A’ — we had to provide a numerical value.
Regression (Figure 2.2) in supervised learning estimates a relationship between dependent and independent variables and provides a real-value score (although the lines are blurred—logistic regression produces a class and real-valued classification produces a real-value). Admittedly, regression is a harder problem but can be a better approach for some datasets [15], [25]. We were constrained to a regression, rather than classification, problem in our context to determine a real-value grade.

2.1.3 Architecture

There are two architectures for scoring sentiment in text: neural networks (typically recurrent neural networks, or RNNs) and lexicons (dictionaries). In brief, neural networks (Figure 2.3) are comprised of interconnected nodes, modeled after the neurons and synapses in the human brain. Real-value vector representations of words in a sentence are fed into the network, which produces a score then compared to the ground truth. The distance from the ground truth is used to tune the node weights (backpropagation). Thus, the network is trained with labeled examples and, over time, learns (but does not communicate) the features that it uses to accurately predict a score. One of the better-known neural networks in sentiment analysis is the Recursive Neural Tensor Network (RNTN) by Socher et al. at Stanford [20]. The network was trained on a Rotten Tomatoes dataset labeled via Amazon
Mechanical Turk and correctly learns various sentence structures as shown in Figure 2.4 (e.g., contrastive conjunctions like “but” and positive or negative negation). The RNTN achieves a high degree of accuracy on sentence-level binary classification (85.4%), but suffers in fine-grained (5-class) accuracy at 45.7%, even though it improved upon the state of the art. Zhang et al. provide a comprehensive survey of various other neural network approaches towards document-, sentence-, and aspect-level classification of sentiment [26].

In contrast, lexicon architectures match weighted or unweighted key words or phrases in a text and aggregate the result into a score. Lexicons may be built manually [27], [28], [9], or by expanding a seed set of words in various ways [29], [30]. Semantic features of a language (e.g., words that negate or intensify sentiment) must be handled by the algorithm. Thus, where neural networks may be thought to learn such rules automatically, lexicon approaches require their explicit implementation. Khoo and Johnkhan provide an analysis of six popular lexicons towards classifying sentiment in news headlines and Amazon product reviews [31].
A notable work joins both architectures by feeding a vector of word sentiment scores (via four lexicon options) into an RNN [32]. In this approach, the authors leverage *semantic compositionality*, or the aggregating of word negators, intensifiers, etc., with *word sense variations*, or the meaning of words in different contexts. While our approach does not currently utilize semantic composition, we do mitigate the effect of word sense variations by building a domain-specific lexicon. Although this work is promising, especially at the two class level (accuracy of 89.2% on the Stanford Sentiment Treebank movie dataset), the five class problem is still fairly inaccurate (accuracy of 51.1% on the same dataset).

Both architectures have their advantages and disadvantages in areas of performance, accuracy, length of text required, and clarity of results. Soroka et al. found that “dictionaries had exceptional precision, but very low recall, suggesting that the method can be accurate,
but that current lexicons are lacking scope. Machine learning systems worked in the opposite manner, exhibiting greater coverage but more error” [33]. Our emphasis on precision, as well as our assumption of unlabeled data, motivated the choice of a lexicon-based approach in addition to its intuitiveness, interpretability (since the rationale for a grade was occasionally requested by students), and accuracy on short segments of text. See [34] for some other limitations of neural networks, especially identifying which linguistic properties are identified and explaining predictions.

2.1.4 Transfer Learning

Transfer learning in the field of machine learning is the process of taking an algorithm trained or tuned in one domain and applying it in another domain. For example, taking the best sentiment classifier in movie reviews and utilizing it in the education domain. From an ethical standpoint, basing student grades on a classifier from a different domain could result in unpredictable and unfair results. However, from a practical standpoint, it has been noted
that sentiment analysis is “highly sensitive to the domain from which the training data is extracted” [15]. Others have found wide disagreement in results when applying the same publicly-available industry sentiment analysis tools to different domains (e.g., accuracy of 80.4%, 62.5%, and 71.5% on movie reviews, tweets, and Amazon reviews, respectively) [35].

Still, various approaches to domain adaptation have been attempted. Most require labeled data from at least the source domain (linking on related features [36] or related words [37]). Others address this task when there is limited labeled data and note performance improvements over ignoring unlabeled data, but still do not address a situation such as ours with no labeled data [38]. Mudinas et al. present a nearly-unsupervised approach based on a two-phase clustering method, noting that a domain-dependent lexicon is still needed or else accuracy will “suffer a serious performance loss once the domain boundary is crossed” [39]. Muhammad et al. generate a hybrid lexicon with weighted values from a local and general-purpose lexicon, but found performance improvements on only two out of three social media datasets with a much lower F1 score on the third [40]. In fact, over half of the combinations they tried decreased the F1 score, even though all lexicons were in the social media domain. Finally, Taboada et al. observe a “clear benefit to creating hand-ranked, fine-grained, multiple-part-of-speech dictionaries for lexicon-based sentiment analysis” [27].

In their paper testing twenty-four algorithms on eighteen datasets (including Twitter, Yelp, Youtube, Amazon, Digg, BBC, Ted, and Myspace) to determine benchmarks in sentiment analysis, Ribeiro et al. note that “sentiment analysis methods cannot be used as ‘off-the-shelf’ methods, especially for novel datasets” and found that “methods are often better in the datasets [in which] they were originally evaluated” [22] even for popular algorithms like SentiStrength [41], [42] and SO-CAL [27]. Haselmayer and Jenny found significant improvement with their crowd-labeled, domain-specific negative word lexicon compared to two other general-purpose lexicons classifying negativity in German media reports and party statements [43]. They, like us, strongly advocate a process and note that even “some
commercial providers advise against using their sentiment lexicon out-of-the-box without customizing it to the domain” [43]. Specifically, when narrowed down to sentiment analysis algorithms in the software engineering domain (Stack Overflow, Jira, etc.), others claim that publicly available sentiment analysis tools are inadequate and disagree with one another, but that domain-specific tools enhance accuracy [44]. Lin et al. even went so far as to label the current state-of-the-art sentiment analysis tools deficient in the software engineering domain [45]. Together, this research shows the necessity of a domain-dependent lexicon and algorithm for accuracy in educational sentiment analysis.

2.2 Sentiment in Education

So far, we have seen options for and examples of sentiment analysis in other domains. In this section, we present different problems that have been solved via sentiment analysis in the education domain. Influential and widely-cited works summarizing sentiment analysis tools and techniques have often lacked any reference to its applications in academia [46], [47], [15]. However, in the past three to four years papers have begun to emerge leveraging sentiment analysis in academia, although to date it has not been applied to generate a student grade based on review comment text. Similarly, we find no research leveraging aspect extraction to semi-automate review form creation in a data-driven way. In both cases, it is important to gather the maximum information possible from available data, as businesses, artists, and politicians have discovered in other domains.

Sentiment analysis has recently been used in the classroom to determine:

- Student attrition over time in three massively open online courses (MOOCs; captured from course forums and scored by a product review lexicon [48]) or predicted attrition in a single MOOC (captured from a course forum and scored by SentiWordNet 3.0 as one feature of a neural network [49])
• The mood of students towards a teacher (captured via Twitter and scored by Naïve Bayes [50]) or as an “emotional thermometer for teaching” in virtual classrooms (captured in forum posts and scored by an ensemble [51])

• Students with a negative outlook or course issues (captured in online course forums and scored by the Microsoft Text Analytics API [52] or captured from social media and scored by a mixed graph of terms [53])

• Teacher strengths and weaknesses identified by students (a proposed system with sentiment captured via questionnaire and scored by Naïve Bayes [54], or a proposed multilingual system with sentiment captured from Coursera peer reviews and scored by a lexicon in R [55], or a system with sentiment captured from teacher evaluations and scored by an ensemble [56])

• Student perception of internship experience (captured from transcribed interviews and scored manually [57])

• An alternative way to view poetry and a means of student discussion on the relationship between text and numbers (captured from a Walt Whitman poem and scored by a “proprietary [sentiment analysis] tool” [58])

While these are all interesting applications, they do not contribute to scalable assessment or the reliability of the grading process. They do not utilize information from students to develop a review form. None go further than classical applications of sentiment analysis — merely noting subjective opinion towards an object.

2.3 Review Form

Davis et al. note that there is no consensus even in a single discipline (biomedical journals) and a narrow focus (research article) for an optimal review form [59]. Of the fourteen
general surgery journals they selected, only two questions were shared among all: overall recommendation and comments to the author. They recommend that a set of guidelines should be created to mitigate “potential gaps [that] exist in the review process” [59]. In this search for a quality review process, the solution is usually pursued in one of two areas: the grading algorithm itself (fairness) and the quality and interpretation of comments (helpfulness). Most study the helpfulness of peer reviews with the understanding that more helpful reviews contribute to more feedback being implemented in future revisions of a work. Rather than research the review form itself, most scholars attempt to teach the student how to review, tweak the software to fix comments, or assign a best-fit reviewer to provide quality feedback.

2.3.1 Determining Helpfulness by Content

Xiong et al. observe two review characteristics (“localization information” and “concrete solutions”) that promote helpful reviews [60]. However, rather than adjust the review form to request the specific characteristics, they propose a natural language processing technique that examines the students’ reviews and prompts them to correct/improve it. Ghose et al. study the impact of subjective content in the related domain of product reviews and find that potential buyers perceived reviews more helpful when they contained a mix of subjective and objective content [61]. Cho and Ipeirotis note the difficulty of an educator monitoring all peer comments as class size grows larger [62]. Their system thus classifies reviewer-tagged comments from three areas of the paper (intro/theory/experimental setup, data analysis/result, and abstract/conclusion) as “helpful” or “non-helpful” based on specificity and praise. The purpose of this study was to monitor comments to filter poor or inappropriate comments before passing them to the student. Our paradigm views the purpose of reviews fundamentally differently than these and other works. We define helpfulness as clarity of review and perception of the work for the instructor, not the student, especially
since research has shown that it is better to give peer review feedback than receive it [12], [63]. Thus, we seek to understand what the student is saying, rather than to fit their review into specific characteristics.

2.3.2 Improving Helpfulness by Matching

Giannoukos et al. focus on peer-matching to improve feedback [64]. Their process involves assigning three to five reviewers based on criteria like proficiency, strictness, usefulness, and willingness to review. Our approach differs in a key way: instead of searching for a few key reviewers (which incidentally, happen to have the profile of an instructor), we seek as many reviewers as possible to gather insights from diversity, rather than conformity. This leaves out no student (who is reviewed by someone with a poor usefulness/willingness/strictness rating) and conversely, prioritizes no student.

2.3.3 Creating Helpfulness by Good Questions

Pechenizkiy et al. note the difficulties of choosing questions for their online assessment that are not too closely related in their small-scale study on data mining student data from a 73-student online exam [65]. While they do not focus on helpfulness per se, they do focus on form revision and employ clustering to determine if answering one question correctly influenced answering another. This analysis could potentially be useful in our iterative process after we collect and revise our review form to prune questions. However, it is sometimes desirable for question overlap, and answering two questions correctly does not signify causality — it may simply indicate a student’s understanding of both sets of material.

Duers describes the learner as co-creator and is perhaps the closest to our idea that students can contribute to the creation of review forms [66]. Her new form, built specifically by twenty-five nursing students, for nursing students, was well received by most but not all students, was condensed, and contained mostly language assessing human qualities like how
“polite”, “professional”, and “responsive” a student was. The study was designed to prevent nurses from feeling “torn to shreds” during peer evaluations. Unlike this study, our review form process is designed to be applicable to any field of study and is based on a corpus of over 10,500 examples of what students actually said. Since it is anonymous peer evaluation, it is not imperative that students hold back or soften their opinion — they can express their true perception of their peers’ presentations to the instructor. This work is perhaps the closest to ours in ideology but differs widely in its breadth and implementation.

2.3.4 Balancing Review Burden and Fairness

Shah et al. propose that peer review alone does not scale since there is a predictable proportion of incorrect peer review scores [1]. In their approach, three to five students review another’s work on a pass/fail basis with the instructor’s grade as the ground truth. Raising the number of reviews per student can become burdensome, so they propose two methods as a form of dimensionality reduction: grouping like submissions which all receive the same grade and grouping like parts of submissions (a method also employed by [67]). In either case, it is difficult to define and assess the clustering algorithm (similarity threshold, max/min number of clusters, etc.) and it seems unfair for a student’s work to receive a score without actually being viewed by a peer. Their educational model also differs from ours, which is a group-specific research project that requires the student to teach their reviewing peers rather than a universal assignment to the entire class.

Kulkarni et al. combine an automated grader and peer review in two pertinent ways: 1) assigning one to three reviewers to a work depending on the confidence of the automated grader or 2) assigning one to three identifiers and one to three verifiers to annotate answer features [4]. The grade is determined by the median of the machine and human (verification) grades. TAs add attributes to the instructor-provided rubric based on the subset of works they graded to determine the ground truth answers.
Identify-verify consumed the same effort as peer-median grading, for 92% of the accuracy in questions with non-binary answers. Fewer students reported liking identify-verify, and students reported low confidence in its grading accuracy, with one verifier citing a concern that other students were not reviewing properly. Only 34% of their submissions had high enough confidence for fewer than three reviewers. They estimated that less than 3% of students (n = 41) who should have passed the course did not due to their grading accuracy (67% to 82%).

The authors only use short answer questions that can be partitioned into components, which are evaluated for presence or absence. This reduces peer reviewing to the monotony of pattern-matching. Additionally, the authors admit to sacrificing grade accuracy (which was highest in peer-median grading) to ease review burden. However, this is not a sacrifice we are willing to make. We desire our students to have confidence in our system. Two aspects of our system lower the reviewing burden of a student: 1) group work (which reduces the number of submissions) and 2) a review-to-learn model with open-ended assignments that introduces variety, requires analysis, and fosters learning.
Chapter 3: Theoretical Framework

“Third-party sellers are kicking our first-party backside. Badly.” – Jeff Bezos

Our theoretical framework was founded on two related ideas: that success could often be found in unlikely places and that a sharing economy (or peer-to-peer economy) with unused capacity could be leveraged with technology to produce unparalleled results, as Amazon CEO Jeff Bezos wrote in his 2018 letter to shareholders [8]. Firstly, we believed students were capable of more than has been credited to them, and secondly, that students were an untapped and valuable resource in the classroom.

Airbnb, Uber, and Amazon (as representative of the housing, transportation, and retail crowd sharing industries) demonstrate the power behind matching demand with supply, particularly when the supply comes from a group of “non-experts” with idle resources [68]. Furthermore, this crowd can make diverse and distributed decisions, the aggregation of which are wise [6]. We wanted to apply this principle to the classroom, with students as the “third-party sellers”, to see how they compared to a “first-party” instructor when challenged to produce and present educational information from their “unused capacity.”

Section 3.1 summarizes the current state of scalable assessment and section 3.2 details how our educational principles are applied within the classroom to fully utilize student resources in a scalable way.

\[2\] Portions of this chapter are from [3] and [9]. Permission is included in Appendix A.
3.1 State of Scalable Assessment

Today there exist a number of online options for students who need or prefer distance learning. Many traditional universities offer partly or completely online programs. A number of online only, massively open online courses (MOOCs) also exist to meet varied educational needs (degree, certification, personal growth, etc.): Coursera, edX, Udacity, MIT OpenCourseWare, and Udemy, among others. Udemy, in particular, employs a crowd-sourced paradigm, allowing anyone to post lectures. Even Youtube could be considered a platform for educational content in this manner. Quality, oversight, and breadth of topics may vary by platform but there is no shortage of resources for online learning. Although teaching lectures online scales trivially, the scalability of grading complex assignments is less obvious.

3.1.1 Assignment Type

In today’s increasingly large and online classrooms, assignments must be provided that maximize the students’ ability to communicate what they know, allow them to express their creativity and independence, encourage critical thinking, and finally, are easy to grade. These constraints initially seem to conflict. Thus, assignments might be chosen not for how they benefit the student, but for ease of grading.

Although open-ended projects are beneficial to students, they are hard to grade in a timely and objective manner in a typical large course [69], [70]. In short, while grading numerical, true/false, and multiple-choice assignments scales easily, grading higher-level open-ended assignments does not. For example, between July 2016 and June 2017 almost 560,000 students took the GRE revised general test, which required responses to two essay prompts, for a total of over one million essays to grade [71]. Similarly, in Dr. Scott Klemmer’s Human-Computer Interaction courses on Coursera, there were five small website creative tasks for each of the 6,700 to 7,200 students to complete [72]. Grading by hand requires an enormous
amount of time and cost. An effective way to grade open-ended assignments is increasingly necessary.

3.1.2 Assessment Options

There are currently two options for assessment of essays in large or massively open online courses: Automated Essay Scorers (AES) or peer review. There are many limitations of AES: missing semantic meaning by focusing solely on textual features, inappropriate (short) content length, susceptibility to gaming, and evaluating factual claims [4], [13]. There are also limitations of peer review including grading time burden and seven common human rater errors: severity/leniency, scale shrinkage, inconsistency, halo effect, stereotyping, perception difference, and rater drift [13]. Zhang postulates that a lack of full understanding of the human rater’s process could bleed into an AES process that is inaccurate [13]. He offers three conditions for fully automated grading: 1) the internal mechanism for grading must be sufficiently transparent 2) enough evidence must be collected to validate fairness and 3) a quality-control mechanism must be available to correct poor results. For a mixture of human and automated assessment, he proposes two options: weighting both or using automated methods as a validation rather than contributing to the rater score.

While AES solves most of the common human rater errors, it does so at the expense of failing to deeply understand the text. It is also confined to the essay domain. In contrast, scaling the number of human raters to take advantage of the wisdom of the crowd averages out individual rater errors and is highly adaptable to other assessment types (e.g., projects). Another useful by-product of a peer-reviewing crowd is the reduction of potential grade bias. Bias exists among graders for multiple reasons: a grader may dislike (or favor) a particular student, be experiencing a particularly frustrating day, or succumb to fatigue. But a peer-reviewing crowd as a whole will not have these limitations as an individual might. Indeed, there may be a few poor reviewers, but their marks will be averaged out.
Thus, we implemented a third option for mixing human and automated grading, leveraging the best of both worlds in a unique way. We utilized the intelligence of peer reviewers to capture content that an AES could not and potentially never will: humor, irony, passion, usefulness, and beauty. We then employed linguistic and natural language processing techniques in the areas in which they excel — concept recognition and sentiment analysis — to assess open-ended projects.

Our model of peer review was both formative and summative — we desired the student to learn by reviewing [73], not to solely review for assessment’s sake so that we could assign a grade at the end of the semester. Additionally, while we acknowledge editorial review (revising and re-submitting work upon receiving feedback) as advantageous to the student [74], our process utilized post-publication peer review so that a student only reviewed and learned from a final submission. This lowered review burden and reduced the transmission of poor quality information.

3.2 Educational Principles

Our research also stemmed from a desire to understand and promote student learning. It is well-known that every student is unique, that every student learns differently, and that engaged students are more successful in a course [75], [76], [77], [78], [79]. Thus, we wanted to customize the learning environment, including the peer review form and assessment process, to their needs.

3.2.1 Learning Environment

In order to employ student-specific content as unused capacity, we created a project-based class of peer teaching with brainstorming, teamwork, and co-creativity. Students learned the material organically, as they would once they graduated [80]. Organic learning (Figure 3.1) is the process of unguided or loosely-guided exploring within the boundaries of a problem
space. According to Bezos, “Wandering is an essential counter-balance to efficiency. You need to employ both. The outsized discoveries – the ‘non-linear’ ones – are highly likely to require wandering” [8].

Organic learning is made highly convenient through the internet — a digital knowledge base that students must learn to wield effectively (see Cummings’ architecture of internet-enabled learning for a more detailed description [81]). Quality information can be found through careful mining, analysis, and problem-solving, then distilled and summarized to be understood by one’s peers. This approach (Figure 3.2) favored all learning styles and allowed creativity and learning at one’s own pace. It also accommodated students who began with different levels of knowledge, allowing gaps in minimum basic knowledge to be filled in as needed.

Organic learning also brings a healthy amount of risk-taking and safe failure into the classroom. “[E]verything needs to scale, including the size of your failed experiments. If the size of your failures isn’t growing, you’re not going to be inventing at a size that can actually move the needle” [8]. There are a number of learning ideologies that attempt to implement failure in the classroom [82], [83], [84], [85]. Although united in their goal (a concept that can be referred to as “graceful failure”, where students realize there are less severe penalties
for making mistakes [86]) each ideology differs on how much and when to lend support to a student. Unfortunately, the traditional educational approach along with standardized testing continues to dominate, producing risk-averse students focused on reciting taught examples to avoid suffering potentially severe consequences. Ultimately, failure will be a part of every student’s life and must be included in the educational arena, with the end goal being, of course, that the student will learn from failure and learn to overcome it [82].

The overall emphasis of our courses was on problem solving and communication, not cramming and then reciting facts. Students gathered data by exerting mental effort and stored it in their working memory as information (see Figure 3.3). They retained course content through organic learning, intelligent summarization, and concise, clear presentation to their peers (a valuable skill in today’s marketplace of ideas). Peers retained course content through peer review. This diminished the atrophy of knowledge through forgetfulness and fostered engagement.

3.2.2 Review Form Questions

Assuming that course projects were chosen well, aligned not only with the desired learning objectives but with students’ backgrounds, interests, and abilities, we faced a challenging task: how should we craft our peer review form? Even in the discipline of engineering, our experience — and to our best knowledge, the only practice — was for the instructor to either
take a pre-packaged review form or to make one up by semi-randomly choosing questions, assigning weights to each question, and summing the scores. We use the term “semi-randomly” because there are rules of thumb to creating a rubric (qualitative vs. quantitative, 5-point Likert scale, analytic vs. holistic, etc.). In reality, though this rubric may weight items according to the instructor’s desires, it may not sample the knowledge field effectively (Figure 3.4). It also may not capture everything the students are saying, and thus its feedback might be incomplete. We desired to probe students’ knowledge using appropriately placed questions that adequately covered the domain. In doing so, we stimulated the students to study as they reviewed. To approach asking the right questions on our review form, we asked the following questions of ourselves:

- How many questions should we ask?
- How should we group the questions?
- How do we weight the questions?
- How should we aggregate the answers?

In addition to the questions, we had a number of considerations to avoid:
3.2.3 Scaling

Our courses and algorithms were developed to facilitate scaling, though we were limited by USF course size and classroom space to approximately 40 students per course. Assuming that limitation was removed, there were a few considerations to scale effectively: we must 1) avoid fatigue from students reviewing too many works and 2) allow all students to participate in presenting to their peers. To accomplish this, we considered semester length, size of groups, and number of groups presenting each week. Our semester length (15 weeks) was constant, but the size of groups (3-4 students) and number of groups presenting each week (1 group) could be allowed to vary for scaling purposes. Since coordination difficulties, lines of communication, and risk of students “skating by” grow with group size, we considered increasing the number of presenting groups the best option. Two or more groups could...
present the same topic to a different subset of students each week, keeping approximately 30-40 peer reviewers per work (Figure 3.5 shows the case when the number of students in a course doubles). Since many of our students preferred learning from the online video or slides posted, this would be as simple as assigning each half of the students to review a different presentation. We use a similar technique to reduce review fatigue for the group essay and term project. In this way, a large online course could be reduced into any number of smaller units — a process much like any divide and conquer algorithm (e.g., merge sort).

![Figure 3.5: Course Scaling](image)

3.2.4 Data

Our peer review data (individual review form responses both analytical and subjective) came primarily from University of South Florida software engineering (SE), software testing (ST), computer graphics (CG), geometric modeling (GM), and data visualization (DV) courses over a period of five semesters. Of the twelve courses, four were undergraduate only, the rest were cross-listed for graduate and undergraduate students.

To date, we have a number of growing corpora of 1) over 10,500 peer reviews from nine SE, ST, CG, and GM courses (over 300 student works), 2) over 2,200 peer reviews from three DV courses (840 student works), and 3) over 2,500 peer reviews from PeerLogic (PL), an open
repository of several peer review systems. These corpora give us a variety of perspectives on the peer review process since each instructor utilizes peer review in different ways.

In the SE, ST, CG, and GM courses, the responses reflected peer sentiment on three different types of projects: a weekly group presentation (35-40 reviewers), a month-long group essay (20-40 reviewers), and a semester-long group term project (20-40 reviewers). In the DV courses, the peer reviews were performed roughly bi-weekly on a semester-long individual visualization programming project by 2-4 reviewers. The PL peer reviews contained no information on the type of project or number of reviewers. Textual responses were aggregated by our system to provide the mean, median, and standard deviation of sentiment, number of comments successfully scored, and various per-comment metrics.
Chapter 4: Aspect Extraction\textsuperscript{3}

“The art of proposing a question must be held of higher value than solving it.”

– Georg Cantor

Rarely, if ever, is a review form subjected to data-driven development or iterative improvement. It is tempting to compile questions after a few moments of thought or to take an existing review form from another source. By faith, we accept that such a review form will accurately communicate information from reviewing students to an instructor, though we have no such evidence. As computer scientists and engineers, we believe that engineering rigor and principles should be applied to the construction of a review form.

In sentiment analysis, aspect extractors are uniquely primed to meet such a need. Aspect extraction is considered a “subtask of sentiment analysis that consists of identifying opinion targets in opinionated text, i.e., in detecting the specific aspects of a product or service the opinion holder is either praising or complaining about” [87]. Aspect extraction is widely used in industry (e.g., Google news pages or Amazon product reviews [88], TripAdvisor hotel reviews [87], and restaurant reviews [89]) to determine sentiment towards objects. In our domain, we determined student sentiment towards specific aspects of their peers’ presentation, essay, or term project. Although our initial review form was created without using an aspect extractor (since some method for determining sentiment in text must be provided first, e.g., relaxation labeling in [90]), we modified and extended our review form in further iterations based on information from our aspect extractor.

\textsuperscript{3}Portions of this chapter are from [9]. Permission is included in Appendix A.
Section 4.1 details the components of and process by which we created our review form and section 4.2 outlines the aspect extractor we built to semi-automate review form revision by gathering information from peer review text.

4.1 Review Form

We began development of our initial review form by combing through peer review text for meaningful content (subsection 4.1.2). Any information-bearing text was a candidate for the review form and was reduced to pertinent words and phrases or question/answer groups (subsection 4.1.3). This form was designed from scratch, based on the students, the courses, and prior teaching experience and style. Since these factors vary widely, our initial form may not be appropriate for other courses. Although the questions chosen were not limited to the field of engineering, they did reflect feedback from students in our discipline. Thus, we do not believe in a one-size-fits-all review form — it is a mistake to use one tuned to a specific context without following the whole process of iteratively mining student comments and updating the review form.

We then used an iterative, data-driven approach to refine the review form [9], which was updated at the end of every semester to probe for the most common student feedback (Figure 4.1). The process was repeated to determine whether the list of questions was acceptable — a steady-state — or if there remained further information to capture. We continued until we found no new topics or key words from student reviews. We periodically renewed the process from a sample of reviews to further validate our existing questions and probe for new topics.

4.1.1 Review Form Components

Our review form was comprised of two components: an analytical section with questions and radio button responses (narrowing, analytic) and a subjective free-response section where
students were encouraged to comment on any aspect of their peers’ work (widening, holistic). Both components were weighted and contributed to a student’s final grade.

Each question/answer group in the analytical section captured a student response in an area that previous students indicated was important (for some examples, see [80]). Questions were then added, modified, or removed after each semester through intelligent data combing (subsection 4.1.2). This process was later semi-automated through the use of an aspect extractor detailed in section 4.2, but still required human intelligence for verification since many aspects were simply too generic.

Selecting a review form radio button response communicated specific, yet limited information. Allowing detailed feedback provided another dimension of student response: sentiment. Though the analytic portion of the review form restricted students, the subjective section allowed complete freedom to discuss anything. Sentiment, along with a concise summary of what students actually said, provided rich information for an instructor to validate a peer review score (or determine exactly why a score was chosen), particularly if the course was large enough that the instructor could not check every work. Thus, the aggregate sentiment
(crowd) score was a valuable component of our final score that could be used to validate or adjust the grade from the analytical section (see chapter 6).

4.1.2 Intelligent Data Combing

Intelligent data combing is the process of selecting information-rich key words and phrases, through human intelligence, to correctly analyze and summarize student observations. In this way, we captured content that other automated graders could not (and perhaps never will): humor, irony, creativity, perceived preparedness, etc. This process was intentionally fuzzy and required human intelligence — words were selected if they provided 1) meaningful sentiment (e.g., “extraordinary” but not “good”) or 2) information (e.g., students mentioned a presence or lack of “citations” or “diagrams”). Initially, aspects were gathered by hand, but later the process was semi-automated (see section 4.2). There were two goals with intelligent data combing: 1) to gather sentiment-bearing words for the lexicon (which is detailed in chapter 5) and 2) to discover aspects — potential question/answer groups to add to the review form. We focus in this chapter on the second goal.

4.1.3 Interpreting Questions

Incentives for peer review in writing courses have been widely studied [91], [92], [93], [94]. In our early courses, reviews were encouraged, but not mandatory. Initially, roughly 75% of the reviews met our basic threshold for scoring with confidence. This number raised to ∼85% when we incentivized students with a point to complete a quality review, which increased the amount of information we received. We noticed student feedback separated naturally into three main clusters of questions: Overall score, Technical score, and Personalization score. Questions and answers were interpreted from the selected words and phrases and placed in the category under which they best fit. Every question on the review form had three possible answers with the exception of Overall score, which had eight. Answers were chosen
to provide the maximum possible semantic distance between choices, e.g. in a question on

*Content*:

- Abundant good quality → A+
- Adequate → B
- Missing a lot → F

The following is one example of a positive review from which we drew concepts that provided questions and answers for our review form:

“This presentation was very well done. The presenters *understood the material* and that was shown in their delivery. The organization of the content was such that it *promoted engagement and triggered discussion*. It was *technically accurate* and provided *a plethora of resources* to be used in the development of the final project. To me, this presentation marks an important milestone (with regards to the information it covers) and I am glad that the presentation enabled a *clear understanding of the material*.”

Thus, we noted or inferred topics that were important to the student — comprehensibility (reviewer or presenter understanding the material), engagement, discussion, soundness (technically accurate), and including resources. We could not simply cut and paste the key words — we had to analyze, summarize, and craft questions and answers like the following on *Comprehensibility*:

- Understood at first reading
- Several readings required
- Incomprehensible
We also took observations from highly negative reviews:

“The only issue is that the presentation was really long. Ridiculously long. After a little more than halfway, it started feeling like a slog — well-written and useful, but a slog nonetheless.”

From this review and others like it, we added questions on creativity and Balance (length):

- Properly balanced
- Rearrangements needed
- Too long/short

4.1.4 Results

Table 4.1 displays the way in which our review form and scoring algorithm were updated. In the first iteration, weighted answers selected by students were summed within their respective section — Overall score, Technical score, and Personalization score — and the section with the least standard deviation (i.e., most agreement) was weighted the highest. In the second iteration, only the grading algorithm changed, weighting sections according to the instructor’s heuristic (e.g., in our case most importance was placed on Technical score). We also increased the granularity of the Overall score from four (excellent, good, marginal, poor) answers to eight (outstanding, very good, good, acceptable, fair, marginal, poor, unacceptable) to increase variance, particularly on mediocre works. The third iteration resulted from removing two questions and adding thirteen others, leveraging our process for capturing aspects. Finally, in the fourth iteration we again modified the grading algorithm. Rather than summing weighted answers within a section, answers were modified to match semantically the weights 0 (worst), 3.0 (average: a ‘B’), and 4.3 (excellent: an ‘A+’) and were averaged within a section. This score was provided in addition to the score by summing (and is typically a few percentage points lower), to provide a full picture to the instructor.
Table 4.1: Review Process Iterations

<table>
<thead>
<tr>
<th>Iteration</th>
<th>Total</th>
<th>Overall</th>
<th>Technical</th>
<th>Personalization</th>
<th>Grading Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>11</td>
<td>1</td>
<td>4</td>
<td>6</td>
<td>Sum within section, weighted by minimum std(^1)</td>
</tr>
<tr>
<td>2</td>
<td>11</td>
<td>1</td>
<td>4</td>
<td>6</td>
<td>Sum within section, weighted by intuition</td>
</tr>
<tr>
<td>3</td>
<td>22</td>
<td>1</td>
<td>16</td>
<td>5</td>
<td>Sum within section, weighted by intuition</td>
</tr>
<tr>
<td>4</td>
<td>22</td>
<td>1</td>
<td>16</td>
<td>5</td>
<td>Average within section, weighted by experiment</td>
</tr>
</tbody>
</table>

\(^1\)standard deviation

4.2 Aspect Extractor

The process of intelligently data combing student text was time-consuming and prone to error (missed content through fatigue, much like grading too many essays in one sitting). Although bootstrapping the review form questions and lexicon by hand could not be avoided (and was in fact desirable for accuracy and intelligibility), we desired a data-driven approach to semi-automate this process and to allow us to more fully understand what students were saying. Aspect extraction, specifically aspect-based sentiment analysis (ABSA) which provided the corresponding sentiment, also afforded several other benefits: we could compare relative number of mentions between different aspects, compare aspects between courses or students (e.g., grads vs. undergrads), or determine general positive/negative sentiment towards each aspect. Finally, aspect extraction could be used to validate the current review form questions, which were all nouns, or to detect “parroting” — students simply mentioning key words from the review form itself.

4.2.1 Design

Our custom aspect extractor was similar to one developed by researchers at Google that identified aspects in restaurant and hotel services [95]. We focused on explicit aspects rather than implicit aspects [96]. We used a sliding window of length seven over sentiment-laden text which suggested aspects from syntactic rules [97]: a regular, comparative, or superlative adjective found in close proximity to a singular or plural common noun, not separated by
punctuation. This was a form of association rule mining between a noun and a set of sentiment words. In Figure 4.2, an adjective (italicized) and noun (highlighted) match is found within the window. The context of and sentiment towards the aspect was saved for later summary. We kept two aspect lists: one for a single group’s work and one for a batch of processed works.

![Figure 4.2: Sliding Window](image)

If an aspect from a batch of works, often an entire semester, met a customizable threshold of mentions and absolute sentiment it was considered a candidate for the review form and displayed to the instructor (Figure 4.3 shows the top aspects from a semester with only one context example instead of the entire list). In the semesters we incentivized students with a point for a quality review, we noticed undergraduate-only courses had a higher variety of aspects (∼294) than mixed courses (∼237).

![Figure 4.3: Example Aspects](image)

In addition to listing all of the review aspects from a batch of works, a summary of the top aspects and their sentiment was provided to the instructor per group submission (Figure 4.4). This showed detailed information on peer review open-ended comments, including the mean
Actual review sentiment mean: 3.76/4.3
Most representative comment(s) scored at 3.75/4.3:

"[The team]'s presentation was very good. They were informative and did a good job pulling information from many sources. They also did a very good job with examples. Examples like the rocket launch failure help to keep the presentation engaging, and their code examples provided real world value. One improvement I would suggest is slides that are a little less dense (text wise). "

Aspect presentation occurs 11 times with sentiment of approximately 0.746 (4.7/6.3):

inspire: "[Their] presentation was inspiring and was a "
useful: "found their presentation useful for future project "
informative: "presentation was really informative . [Student] seems "
knowledgeable: "presentation and very knowledgeable about their subject "
engag: "the presentation more engaging . Having specific "
short: "presentation was cut short the class discussion "

Aspect job occurs 4 times with sentiment of approximately 1.0 (3.9/3.9):

fantastic: "they did a fantastic job presenting . "
outstand: "did in a outstanding job at showing "
phenomenal: "[Team] did a phenomenal job on their "
extcellent: "[Team] did an excellent job expanding on "

Aspect touch occurs 3 times with sentiment of approximately 1.0 (0.9/0.9):

nice: "was a real nice touch to end "
nice: "images were a nice touch to also "
nice: "videos were a nice touch and helped "

Figure 4.4: Aspect Summary for a Single Work

of all comments, a representative comment with the closest score to the mean, and the three most frequently occurring aspects with up to six key sentiment words each. For each aspect, the sentiment score is scaled from negative one to one. The ratio of positive to negative sentiment words displayed is adjusted to match the sentiment score. Finally, the sentiment words provided are the top positive and negative weighted key words that correspond to each aspect. Although we had more comprehensive visuals for an instructor to analyze a peer review, this single page summary conveyed the overall sentiment concisely and accurately.
4.2.2 Validation

4.2.2.1 External

To validate our aspect generator, we used a data set from a well-known contest: SemEval-2016 Task 5: Aspect-Based Sentiment Analysis. We focused on Subtask1 (SB1) of their restaurant data set, which was comprised of English reviews manually tagged with aspect and sentiment (positive, negative, or neutral). Our aspect extractor only found aspects in proximity to opinionated text. This allowed us to determine multiple sentiments on a single aspect but not aspects with no sentiment, thus we removed those with neutral sentiment. After scraping and cleaning the data, the training data set had 1,312 aspects and the testing data set had 460 aspects. Sentences could contain more than one aspect.

One focus of our aspect extractor was to group aspects and their corresponding opinions as a summary from a selection of reviews. When evaluating precision, recall, and F1 score we had two options: 1) a sentence-by-sentence basis (non-grouping — array-based) or 2) an overall basis (grouping — hash-based and summing sentiment). In both instances, since our aspect extractor produced a fine-grained, continuous sentiment score, we rounded to either 1 or -1 to compare to the test set ground truth. In general, the precision by grouping was lower (since different opinions were expressed on the same aspect by different reviewers) but recall was higher (since an aspect missed in a particular review could be found later). Grouping (or aspect-based sentiment summarization) is an important tool for many online systems today [95]. In our case, precision was more important for grading text (chapter 6) and recall was more important for finding potential aspects to add to the review form.

To establish a baseline we processed the reviews using our aspect extractor on the test set without any training, using our current lexicon. The result is shown in the first row of Table 4.2. The precision was better than the top entry (88.126) of 168 submissions from 29 teams (Slot3, Acc in [98]). We then added two positive key words and one negative key
Table 4.2: Restaurant Aspect Extractor

<table>
<thead>
<tr>
<th></th>
<th>Non Grouping</th>
<th></th>
<th></th>
<th>Grouping</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Precision</td>
<td>Recall</td>
<td>F1</td>
<td>Precision</td>
<td>Recall</td>
<td>F1</td>
</tr>
<tr>
<td>Baseline</td>
<td>0.918</td>
<td>0.141</td>
<td>0.244</td>
<td>0.875</td>
<td>0.222</td>
<td>0.354</td>
</tr>
<tr>
<td>Good, great, bad</td>
<td>0.936</td>
<td>0.254</td>
<td>0.400</td>
<td>0.851</td>
<td>0.348</td>
<td>0.494</td>
</tr>
<tr>
<td>With learning</td>
<td>0.923</td>
<td>0.330</td>
<td>0.486</td>
<td>0.860</td>
<td>0.422</td>
<td>0.566</td>
</tr>
</tbody>
</table>

word — *good, great, and bad* — to increase the recall (second row of Table 4.2), since we intentionally excluded such widely-utilized words from our lexicon. Finally, we found and tagged the polarity of key adjectives (intelligent data combing) found in the training data set (third row of Table 4.2). This was a form of learning (adding 104 positive and 92 negative words) and also of adapting our lexicon to the restaurant domain (e.g., including words like *bitter, bland, expensive, delicious, fresh,* and *organic* that make no sense when reviewing a group’s presentation). Increasing the lexicon size doubled recall from the baseline for non-grouped aspects. When simply searching for targets, our aspect extractor places 6th in constrained (domain-limited) submissions when grouping, 14th overall (Slot2 in [98]). It is also interesting to note that our aspect extractor found some aspects missed by those tagging the reviews (Table 4.3).

4.2.2.2 Internal

Of equal importance to external validation was internal validation. Since the aspect extractor was developed in a specific course context, it should assist with finding aspects important to the instructor in that context. To test this, the peer review text for two student works was manually tagged by the instructor, who did not participate in the development of the aspect extractor. The marked-up text consisted of words and phrases denoted either positive or negative. Rather than annotate repetitions of important concepts, the instructor marked the first instance of each (grouping). Annotations were then coded by their explicit
Table 4.3: Restaurant Aspects not Previously Found

<table>
<thead>
<tr>
<th>Aspect</th>
<th>Sentence</th>
</tr>
</thead>
<tbody>
<tr>
<td>place</td>
<td>This is a great place to get a delicious meal.</td>
</tr>
<tr>
<td>location</td>
<td>It’s located in a strip mall near the Beverly Center, not the greatest location, but the food keeps me coming back for more.</td>
</tr>
<tr>
<td>experience</td>
<td>We have since returned and also had a great experience, sampling more small plates and a variety of the beer (cold and good).</td>
</tr>
<tr>
<td>service</td>
<td>We did have the same waiter the second time, so maybe the service is spotty and our luck is good.</td>
</tr>
<tr>
<td>fruit</td>
<td>Eggs, pancakes, potatoes, fresh fruit and yogurt – everything they serve is delicious.</td>
</tr>
<tr>
<td>fish</td>
<td>Great meal - the fish on the omikase platter was absolutely decadent – there was none of the stringiness that sometimes accompanies fair sushi – this fish was perfect!!!!</td>
</tr>
<tr>
<td>yellowtail</td>
<td>Try the Chef’s Choice for sushi as the smoked yellowtail was incredible and the rolls were also tasty.</td>
</tr>
<tr>
<td>waterfront</td>
<td>Great seasonal fish and seafood, with a classy waterfront setting.</td>
</tr>
<tr>
<td>sushi</td>
<td>If you’re interested in good tasting (without the fish taste or smell), large portions and creative sushi dishes this is your place...</td>
</tr>
<tr>
<td>server</td>
<td>To start things off, our lovely server Brooke was quickly on hand to take my drink order.</td>
</tr>
</tbody>
</table>

or implicit aspect and polarity. Implicit aspects not containing a noun, e.g., teamwork in “worked well together” or uniqueness in “has not stood out as clearly different”, were not included, since our aspect extractor did not infer aspects.

In the mixed graduate and undergraduate work, the aspect extractor found 90% of aspects coded by the instructor (n = 21), and of those, classified the polarity of 93% correctly. In the undergraduate only work, the aspect extractor found 67% of aspects coded by the instructor (n = 21), and of those, classified the polarity of 79% correctly. Both times the aspect extractor found additional valid aspects like imagery, figures, video, and depth. Because our aspect extractor focused only on aspects (italicized) in proximity to sentiment (underlined), there were several cases discovered in which potential aspects could be missed:

1. Aspects greater than four words (inclusive) away from sentiment, e.g., “the length of the presentation was just right.”
2. Aspects next to noisy key words intentionally excluded from the lexicon, e.g., “great amount of detail.”

3. Aspects next to negated negative key words (which received neutral sentiment in our scheme and would not be counted), e.g., “rendering with mosaic concepts should have been explained in more detail, it was more like a read out.”

4. Aspects next to a statement of fact (or no inherent sentiment words), e.g., “in my opinion evenly distributed workload is vital.”

We found the level of recall in these two instances reasonably acceptable. The presence of inferred/implicit aspects underlies the importance of including intelligent data combing, rather than relying solely on the aspect extractor.
Chapter 5: Lexicon

“Gracious words are like a honeycomb, sweetness to the soul and health to the body.” – Proverbs 16:24

Any system that attempts to understand text must balance an inverse relationship between breadth of coverage and depth of understanding. This dichotomy is analogous to a choice between recall and precision in machine learning. In building our domain-dependent lexicons, we prioritized depth of insight and precision over breadth and recall since our system produced a grade (i.e., high stakes). While we required a certain level of coverage to ensure confidence in assigning a comment grade per review, we intentionally excluded over-used words that served as noise rather than providing quality information. The sizable number of peer reviewers (typically 25-40) permitted us to be more selective and handle some comments with fewer key word matches.

Even within our single discipline of computer science and engineering, we found it necessary to modify our lexicons in certain courses to maintain contextual polarity [100]. In Software Testing, for example, bug and fix were often not negative words as in general vernacular. Instead, they typically indicated that a team had successfully found and corrected a seeded fault (i.e., a positive sentiment). Thus, we reiterate the sentiment of previous work (chapter 2) that the process, rather than the lexicons, should be copied.

Section 5.1 outlines the process of key word selection and describes our lexicon, HeLPS: the Heuristic Lexicon of Peer Sentiment [101]. Section 5.2 details the polarity (direction and weight) of key words. In section 5.3 we list and describe other publicly-available lexicons.

4Portions of this chapter are from [99]. Permission is included in Appendix A.
Finally, section 5.4 presents experiments contrasting the usefulness and precision of different lexicons.

5.1 Process

Over the course of five semesters (nine courses), we gathered sentiment-bearing key words for HeLPS through the process of intelligent data combing described in chapter 4 (similar to the concept-based approach leveraging a human knowledge base in [47]). In contrast to the review form questions, which were selected for their breadth, we selected any words (even those used very infrequently) that exhibited positive or negative sentiment — others have also found a balance of frequent and rare words necessary to discover subjective content [102]. Human intelligence was required for this task — we could not simply select the most common feedback (e.g., good or bad) because it did not add meaningful information. Instead, we cut through the noise and selected only unique words and phrases that provided rich meaning. Table 5.1 shows some sample lexicon key words.

Key words were then stemmed and weighted by instructor heuristic (section 5.2). Thus, HeLPS could be interpreted as a stemmed seed set which was not expanded through a lexical learning strategy since we desired a smaller set of words specific to our domain and weighted by heuristic (in contrast to [29], [96], and [103]). Our positive word list currently contains 283 words and our negative word list currently contains 207 words. We have an additional list comprised of words that negate sentiment (22 words) and a list for flag words like cheating.

<table>
<thead>
<tr>
<th>Positive</th>
<th>Negative</th>
<th>Negate</th>
<th>Flag</th>
</tr>
</thead>
<tbody>
<tr>
<td>teamwork</td>
<td>lag</td>
<td>not</td>
<td>copy</td>
</tr>
<tr>
<td>fascinate</td>
<td>slow</td>
<td>cannot</td>
<td>paste</td>
</tr>
<tr>
<td>innovative</td>
<td>heavy</td>
<td>no</td>
<td>infringement</td>
</tr>
<tr>
<td>in-depth</td>
<td>improve</td>
<td>none</td>
<td>cheat</td>
</tr>
<tr>
<td>nicely</td>
<td>lost</td>
<td>nothing</td>
<td>cheater</td>
</tr>
</tbody>
</table>

Table 5.1: Sample of Lexicon Words
The inclusion of a flag word list allowed us to crowdsource plagiarism detection. Our lexicon was much smaller than the next closest out-of-domain lexicon we compared in section 5.3. However, because the words were chosen in-domain, the information generated compared favorably to other, larger lexicons.

We tracked the variety of key words used per work and per semester, as well as the percentage of lexicon matched per work (typically 18-20% of the positive list and 4-8% of the negative list). Figure 5.1 shows a word cloud from WordArt.com of a full semester of mixed positive and negative key words from a Software Engineering course. Word size correlates to the number of mentions. The top key word was *example*, which was recorded 9,809 times. The least-mentioned key words were *facilitate*, *character*, *excite*, *logical*, and *trivial*, which were all mentioned twice. Figure 5.1 exemplifies both general (e.g., *enjoy* and *useful*) as well as domain-specific key words (e.g., *robust* and *diagram*).

## 5.2 Polarity

The polarity of a word, phrase, or sentence is comprised of direction (positive, negative, neutral) and an optional weight. In subsection 5.4.1 we compare weighted and unweighted lexicon variants.

### 5.2.1 Direction

To classify the direction of key words found during intelligent data combing, we grouped tokens (words and punctuation) into six categories based on context within the sentence: positive word, negative word, neutral word, negate word, flag word, and reset token. Figure 5.2 demonstrates how each category fits into three sentiment directions — positive, negative, neutral. Flag words (e.g., *copying*) were always negative and reset tokens were always neutral (e.g., *however*). Some negate words were negative (e.g., *missing*) and some
Figure 5.1: Word Cloud of Semester Key Words

were neutral (e.g., not). The particular strategy we used to negate sentiment when grading is discussed in detail in chapter 6.

5.2.2 Weight

After the direction of sentiment was determined, key words were weighted by instructor heuristic as opposed to a learning strategy [104], [105]. Other lexicons are similarly weighted by an expert [28], small group [106], or a crowd (e.g., Amazon Mechanic Turk workers in [43], [107], and [108]). For simplicity, both positive and negative key words were weighted on a continuous scale between zero and one.
5.3 Other Lexicons

A wide variety of lexicons for sentiment analysis currently exist. A handful of lexicon/software combinations are proprietary and for licensed use only (e.g., WordStat and Linguistic Inquiry and Word Count (LIWC)) but many are available for academic purposes. The two primary formats for lexicons are as follows:

5.3.1 Synsets

A synset is a semantically-related set of words — *cognitive synonyms* — that share meaning. A synset is comprised of an ID that uniquely identifies the row as well as a positive value, a negative value, (potentially) an objectivity value (1 - (Pos-value + Neg-value)), a gloss which includes one or more words that share a meaning, a definition, and one or more examples. The fundamental thought is that polarity can be found by examining related words — positive words are connected to other positive words and vice-versa. An example synset is as follows:

\{02084101 0.5 0.25 studious#2 bookish#1 characterized by diligent study and fondness for reading; “a bookish farmer who always had a book in his pocket”; “a quiet studious child”\}

SentiWordNet 3.0 [105] and Micro-WNOp are examples of synset lexicons that provide sentiment scores to synset entries from Princeton’s WordNet, an unlabeled lexicon.
5.3.2 Word-Value Pair

In contrast to synsets, word-value pair lexicons do not store information about other semantically-related words and assume that a word has either an exclusively positive or negative value. This simplifies the calculation of sentiment and negation, although it can be argued that it does forgo some contextual elasticity since a particular word can be used in different ways. To limit the impact of choosing an exclusive sentiment direction, lexicons are often domain-specific. Examples of such lexicons include Affective Norms for English Words (ANEW) [106], SlangSD [104], Multi-Perspective Question Answering (MPQA) [109], Valence Aware Dictionary and sEntiment Reasoner (Vader) [108], and AFINN-111 [28]. Interestingly, SlangSD incorporates slang words scraped from UrbanDictionary.com and Vader includes emoticons (e.g., :D), word shape (e.g., all caps to denote passion), punctuation that increases intensity (e.g., !!!), and acronyms (e.g., lol). For an academic domain such as student peer reviews we have not noticed situations in which the additional breadth of these lexicons is necessary, however. Finally, MPQA includes misspelled words that the authors found frequently occurred.

Due to their simplicity, domain-centric focus, and intuitive design we chose to implement HeLPS as the latter class of lexicons, the word-value pair. Any lexicons in synset format were converted to word-value pair format by extracting only words with an exclusively positive or negative sentiment.

5.3.3 Formatting and Cleaning

Each lexicon compared in section 5.4 had slight differences that required adaptation to match our format. For every lexicon that included a numeric value for sentiment, we produced both a weighted and unweighted variant. When mapping from a different scale to our scale of [0, 1] we mapped [0.1, 1] so that the least-weighted positive and negative
words would not receive a weight of 0. Finally, none of the other lexicons were stemmed, thus comments were not stemmed for comparison. The descriptions and formatting of each lexicon we tested are as follows:

- **AFINN-111**: manually labeled by Finn Arup Nielsen in 2009-2011. Words were originally scored on a scale of [-5, 5]. Only words with a sentiment [-5, -2] and [2, 5] were kept. The resulting tagged word set was thus reduced from 2,477 to 670 positive and 1,289 negative words.

- **ANEW-2017**: manually labeled by a group of introductory psychology students for class credit. Words were originally scored on a scale of [0, 9]. Only words with a sentiment [0, 3] and [6, 9] were kept. The resulting tagged word set was thus reduced from 3,189 to 1,162 positive and 436 negative words.

- **MPQA**: automatically labeled by using a seed set with provided polarity orientations expanded through a WordNet search for semantically related words. Unlike the other lexicons, words did not receive a fine-grained score. Thus, the word sets did not have to be mapped numerically and only an unweighted lexicon was extracted. The number of tagged words remained the same at 2,007 positive and 4,783 negative words.

- **SentiWordNet 3.0**: automatically labeled similarly to MPQA, using a semi-supervised approach with WordNet. Words were originally scored on a scale of [0, 1] and thus were not scaled. Only words with an exclusive sentiment (positive or negative) in the range of [0.1, 1] were kept. The number of tagged words was thus reduced from 117,659 to 9,069 positive and 9,601 negative words.

- **SlangSD**: automatically labeled 1) via existing lexicons — SentiWordNet, LIWC, MPQA, and a previous work (1% of the lexicon) 2) leveraging Twitter (average of the nearest words in 150 tweets containing the target word, 23% of the lexicon), and 3) from a
seed set of related words expanded on UrbanDictionary (76%). Words were originally scored on a scale of [-2, 2]. Only words with a sentiment [-2, -1] and [1, 2] and of length one were kept. The resulting tagged word set was thus reduced from 96,461 to 10,479 positive and 29,025 negative words.

- Vader: manually labeled by ten independent raters. Words were originally scored on a scale of [-4, 4]. Only words with a sentiment [-4, -2] and [2, 4] were kept. The resulting tagged word set was thus reduced from 7,517 to 1,009 positive and 1,237 negative words.

It is interesting to note that, with the exception of HeLPS and ANEW, the negative list length surpassed that of the positive list in all lexicons.

5.4 Experiment

It is difficult to quantitatively evaluate a lexicon’s contents to an external standard. The inclusion or exclusion of a particular word (and its corresponding weight) is innately qualitative. We could, however, contrast the aggregation of sentiment between different lexicons by holding the scoring algorithm constant (covered and evaluated in chapter 6). Although we had no ground truth for the review text, we assumed that if the coverage of a lexicon was adequate the aggregate sentiment score would be in the general proximity (e.g., same letter grade) of the aggregate review form analytical score. Since boundary cases result in different letter grades, we primarily evaluated the mean absolute error between the aggregate form score and the mean or median sentiment score. Note that we did not compare a single student’s comment to their corresponding review form analytical score, rather, the aggregation of each from 25-40 reviews.

We define the following attributes:
• FormScore (FS) is the mean of a collection of reviewers’ radio button responses towards a single student work, with the Overall section weighted 0.3, Technical section weighted 0.4, and Personalization section weighted 0.3: \([0, 4.3]\)

• MeanSentiment (MS) is the mean of a collection of review comment sentiment scores towards a single student work: \([0, 4.3]\)

• MedianSentiment (MdS) is the median of a collection of review comment sentiment scores towards a single student work: \([0, 4.3]\)

• AvgMatch (AM) is the average percentage of student works (36 per semester) where FormScore has the same letter grade as MeanSentiment: \([0, 1]\)

• MeanAbsError (MAE) is the absolute difference between FormScore and MeanSentiment, an accumulation of error in works over a given course: \([0, \infty)\)

• MedianAbsError (MdAE) is the absolute difference between FormScore and MedianSentiment, an accumulation of error in works over a given course: \([0, \infty)\)

We evaluated the above attributes for four recently completed courses: Computer Graphics (CG), Software Testing (ST), Geometric Modeling (GM), and Software Engineering (SE). ST and SE were undergraduate only, CG and GM were mixed undergraduate and graduate. Each course had approximately 35-45 students per semester.

5.4.1 Weighted or Unweighted

Intuitively, an intelligently-weighted lexicon should outperform an unweighted one. Similarly, a lexicon weighted by an instructor should more closely match their desire when processing text. Table 5.2 demonstrates a comparison between some key positive attributes when using the weighted vs. unweighted lexicons: lower standard deviation, lower mean absolute error, lower median absolute error, and higher average matched. For each metric, with
Table 5.2: Weighted Versus Unweighted Lexicon

<table>
<thead>
<tr>
<th>Course</th>
<th>LowerStddev</th>
<th>LowerMAE</th>
<th>LowerMdAE</th>
<th>HigherAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>CG</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>W</td>
</tr>
<tr>
<td>GM</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>W</td>
</tr>
<tr>
<td>ST</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>U</td>
</tr>
<tr>
<td>SE</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>U</td>
</tr>
</tbody>
</table>

the exception of a higher average match for ST and SE, the weighted versions were found superior. Standard deviation fell by 49-56%, mean absolute error was reduced by 57-89%, and median absolute error lowered by 45-72%. It should also be noted that the unweighted versions had a higher average mean (1-4%) and median (3-5%) than the weighted versions. Although the unweighted versions were slightly more generous, that did not in itself indicate greater correctness.

5.4.2 Matching Form Score

Since the weighted lexicons as a group were found to have superior general qualities, we used them to compare between lexicons. Table 5.3 shows the lexicons mean/median absolute error and average matched, with the best scores bolded. In three of the four courses, HeLPS was first in a majority of metrics. Most importantly, we found the lowest mean absolute error for all courses but ST. ANEW appeared to be the next most accurate lexicon, performing particularly well on ST. The two largest lexicons, SentiWordNet and SlangSD, performed significantly worse than the others. Finally, in general, the lexicons captured content in the mixed graduate and undergraduate courses better. HeLPS matched 75-77% of the time rather than 28-42% of the time, which was surprising considering that it was created from content in all four courses.
Table 5.3: Weighted Lexicon Comparison

<table>
<thead>
<tr>
<th>Lexicon</th>
<th>CG MAE</th>
<th>MDAE</th>
<th>AM</th>
<th>GM MAE</th>
<th>MDAE</th>
<th>AM</th>
<th>ST MAE</th>
<th>MDAE</th>
<th>AM</th>
<th>SE MAE</th>
<th>MDAE</th>
<th>AM</th>
</tr>
</thead>
<tbody>
<tr>
<td>HeLPS_W</td>
<td>0.115</td>
<td>0.167</td>
<td>0.771</td>
<td>0.083</td>
<td>0.132</td>
<td>0.750</td>
<td>0.209</td>
<td>0.148</td>
<td>0.389</td>
<td>0.207</td>
<td>0.169</td>
<td>0.250</td>
</tr>
<tr>
<td>ANEW_W</td>
<td>0.165</td>
<td>0.159</td>
<td>0.514</td>
<td>0.150</td>
<td>0.147</td>
<td>0.417</td>
<td>0.235</td>
<td>0.147</td>
<td>0.417</td>
<td>0.251</td>
<td>0.222</td>
<td>0.250</td>
</tr>
<tr>
<td>VADER_W</td>
<td>0.170</td>
<td>0.157</td>
<td>0.543</td>
<td>0.150</td>
<td>0.149</td>
<td>0.417</td>
<td>0.313</td>
<td>0.266</td>
<td>0.083</td>
<td>0.363</td>
<td>0.351</td>
<td>0.083</td>
</tr>
<tr>
<td>AFINN_W</td>
<td>0.228</td>
<td>0.210</td>
<td>0.571</td>
<td>0.237</td>
<td>0.231</td>
<td>0.472</td>
<td>0.343</td>
<td>0.286</td>
<td>0.083</td>
<td>0.384</td>
<td>0.375</td>
<td>0.083</td>
</tr>
<tr>
<td>SWN_W</td>
<td>0.385</td>
<td>0.370</td>
<td>0.200</td>
<td>0.431</td>
<td>0.431</td>
<td>0.111</td>
<td>0.517</td>
<td>0.486</td>
<td>0.028</td>
<td>0.530</td>
<td>0.521</td>
<td>0.111</td>
</tr>
<tr>
<td>SlangSD_W</td>
<td>0.574</td>
<td>0.550</td>
<td>0.171</td>
<td>0.591</td>
<td>0.580</td>
<td>0.083</td>
<td>0.634</td>
<td>0.604</td>
<td>0.028</td>
<td>0.685</td>
<td>0.674</td>
<td>0.083</td>
</tr>
</tbody>
</table>

Table 5.4: Lexicon Information from Graduate Courses

<table>
<thead>
<tr>
<th>Lexicon</th>
<th>PosWords</th>
<th>NegWords</th>
<th>PosSenti</th>
<th>NegSenti</th>
<th>PosWords</th>
<th>NegWords</th>
<th>PosSenti</th>
<th>NegSenti</th>
</tr>
</thead>
<tbody>
<tr>
<td>HeLPS_W</td>
<td>50.7</td>
<td>7.9</td>
<td>2.345</td>
<td>-0.461</td>
<td>50.4</td>
<td>8.5</td>
<td>2.751</td>
<td>-0.550</td>
</tr>
<tr>
<td>ANEW_W</td>
<td>62.7</td>
<td>2.6</td>
<td>2.104</td>
<td>-0.153</td>
<td>69.7</td>
<td>3.9</td>
<td>2.232</td>
<td>-0.229</td>
</tr>
<tr>
<td>VADER_W</td>
<td>30.3</td>
<td>4.9</td>
<td>0.775</td>
<td>-0.061</td>
<td>39.4</td>
<td>9.9</td>
<td>0.980</td>
<td>-0.142</td>
</tr>
<tr>
<td>AFINN_W</td>
<td>32.8</td>
<td>7.7</td>
<td>0.937</td>
<td>-0.077</td>
<td>38.2</td>
<td>12.9</td>
<td>1.188</td>
<td>-0.203</td>
</tr>
<tr>
<td>SWN_W</td>
<td>281.1</td>
<td>76.8</td>
<td>2.437</td>
<td>-1.583</td>
<td>362.8</td>
<td>105.6</td>
<td>2.726</td>
<td>-1.836</td>
</tr>
<tr>
<td>SlangSD_W</td>
<td>220.1</td>
<td>145.1</td>
<td>0.325</td>
<td>-0.651</td>
<td>314.4</td>
<td>261.2</td>
<td>0.525</td>
<td>-0.731</td>
</tr>
<tr>
<td>HeLPS *</td>
<td>*</td>
<td>*</td>
<td>3.332</td>
<td>-0.817</td>
<td>*</td>
<td>*</td>
<td>3.776</td>
<td>-0.903</td>
</tr>
<tr>
<td>ANEW *</td>
<td>*</td>
<td>*</td>
<td>4.421</td>
<td>-0.327</td>
<td>*</td>
<td>*</td>
<td>4.435</td>
<td>-0.432</td>
</tr>
<tr>
<td>AFINN *</td>
<td>*</td>
<td>*</td>
<td>2.474</td>
<td>-0.483</td>
<td>*</td>
<td>*</td>
<td>2.706</td>
<td>-0.675</td>
</tr>
<tr>
<td>SWN *</td>
<td>*</td>
<td>*</td>
<td>6.915</td>
<td>-5.112</td>
<td>*</td>
<td>*</td>
<td>7.635</td>
<td>-5.863</td>
</tr>
<tr>
<td>SlangSD *</td>
<td>*</td>
<td>*</td>
<td>0.954</td>
<td>-4.289</td>
<td>*</td>
<td>*</td>
<td>1.350</td>
<td>-4.497</td>
</tr>
<tr>
<td>MPQA *</td>
<td>40.1</td>
<td>19.1</td>
<td>0.251</td>
<td>-0.043</td>
<td>60.2</td>
<td>38.3</td>
<td>0.430</td>
<td>-0.108</td>
</tr>
</tbody>
</table>

* same as weighted version

5.4.3 Lexicon Information

Since the size of the lexicons varied widely, it was of some interest to compare the average lexicon words matched per work and average sentiment discovered per review to check the information received. Table 5.4 shows the average 1) lexicon words matched from 25-40 reviews on a single work (Pos/NegWords) and 2) sentiment of key words per review for CG and GM. Table 5.5 references the same information for ST and SE.

In the weighted versions, HeLPS, ANEW, and SentiWordNet collected the top positive and negative sentiment. HeLPS compared favorably with SentiWordNet even though our lists contained just 2% and 3% of SentiWordNet’s negative and positive lists, respectively. ANEW, a hand-scored lexicon, generally found roughly the same positive sentiment, but less negative sentiment than our lexicon lists (which were 47% and 24% smaller, respectively).
Table 5.5: Lexicon Information from Undergraduate Courses

<table>
<thead>
<tr>
<th>Lexicon</th>
<th>ST</th>
<th>SE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PosWords</td>
<td>NegWords</td>
</tr>
<tr>
<td>HeLPS</td>
<td>40.5</td>
<td>12.8</td>
</tr>
<tr>
<td>ANEW_W</td>
<td>62.7</td>
<td>6.1</td>
</tr>
<tr>
<td>VADER_W</td>
<td>32.3</td>
<td>11.1</td>
</tr>
<tr>
<td>AFINN_W</td>
<td>34.8</td>
<td>16.8</td>
</tr>
<tr>
<td>SWN_W</td>
<td>308.3</td>
<td>115.2</td>
</tr>
<tr>
<td>SlangSD_W</td>
<td>251.5</td>
<td>232.2</td>
</tr>
<tr>
<td>HeLPS</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>ANEW</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>AFINN</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>SWN</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>SlangSD</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>MPQA</td>
<td>48.2</td>
<td>38.3</td>
</tr>
</tbody>
</table>

* same as weighted version

As expected, the unweighted lexicons score higher than their weighted counterparts, since weighting reduced polarity. Of all the lexicons, only SlangSD appeared to be better at finding negative sentiment than positive.

5.4.4 Qualitative Example

While it is true that a lexicon must find enough sentiment for confidence in a grade, the accuracy of the sentiment must also be maintained. Figure 5.3 provides one qualitative example of the three top sentiment-producing lexicons on a single review from a ST group presentation on “How failures come to be”. The review highlights differences in how the lexicons interpreted text.

SentiWordNet matched the most words, but not in an intuitive way (e.g., very as positive or only as negative). ANEW’s matching was more intuitive, perhaps because its words were selected and weighted by human intelligence, like ours. Ultimately, while the other lexicons found and classified words that are typically negative (e.g., traffic, problem, failure, and infection — which SentiWordNet surprisingly classified as positive), our domain-dependent words intentionally excluded from the lexicon were correctly labeled as neutral.
When coupled with the results of Table 5.3, it becomes clear that simply matching the most key words does not guarantee success. We observed that quality trumps quantity, suggesting that sometimes less is more. We found that HeLPS performed precisely and accurately in our domain especially when compared to other publicly-available automatic- or hand-ranked lexicons. HeLPS resulted in the lowest difference between aggregate review form analytical score and aggregate comment score and consistently tagged high-quality positive and negative sentiment with a lexicon a fraction of the size of others.
HeLPS_W
Grade: 3.93/4.00 (A)
I really enjoyed the presentation. It was very informative and the team was well prepared. I liked how they started off with how the term "bug" came about. I also agree with professor Piegl, I have seen code that is messier than the example they presented, but I liked how the made the connection between bugs and messy code. I learned the "traffic" mnemonic, which are the seven steps of debugging: Track the problem, Reproduce the failure, Automate and simplify test cases, Find origins of infection, Focus on most likely causes, Isolate the infection chain, and Correct the defect. And I also liked the name of their company "Bugs Busters", very original. I only wish some of the members would have speak up a little bit louder, but overall, good presentation. I really enjoyed it.

ANEW_W
Grade: 3.16/4.00 (B)
I really enjoyed the presentation. It was very informative and the team was well prepared. I liked how they started off with how the term "bug" came about. I also agree with professor Piegl, I have seen code that is messier than the example they presented, but I liked how the made the connection between bugs and messy code. I learned the "traffic" mnemonic, which are the seven steps of debugging: Track the problem, Reproduce the failure, Automate and simplify test cases, Find origins of infection, Focus on most likely causes, Isolate the infection chain, and Correct the defect. And I also liked the name of their company "Bugs Busters", very original. I only wish some of the members would have speak up a little bit louder, but overall, good presentation. I really enjoyed it.

SentiWordNet_W
Grade: 3.36/4.00 (B+)
I really enjoyed the presentation. It was very informative and the team was well prepared. I liked how they started off with how the term "bug" came about. I also agree with professor Piegl, I have seen code that is messier than the example they presented, but I liked how the made the connection between bugs and messy code. I learned the "traffic" mnemonic, which are the seven steps of debugging: Track the problem, Reproduce the failure, Automate and simplify test cases, Find origins of infection, Focus on most likely causes, Isolate the infection chain, and Correct the defect. And I also liked the name of their company "Bugs Busters", very original. I only wish some of the members would have speak up a little bit louder, but overall, good presentation. I really enjoyed it.

Figure 5.3: Qualitative Comparison of Lexicons
Chapter 6: Sentiment Analysis

“An investment in knowledge pays the best interest.” – Benjamin Franklin

Integral to our process of scalable and reliable assessment was our sentiment analysis algorithm, SentiSoft. Without an intelligent and transparent process to aggregate first individual words then full reviews, the best lexicon would have been useless and the results unexplainable to a student inquiry. Our algorithm, like our lexicon, incurred a number of updates over time (see Table 4.1 for details) and currently provides an ensemble of scores from which the instructor can infer a reliable grade.

It is important to note that SentiSoft is not intended to be a fully-automated system from which the instructor can generate a grade and move on with their life. The human element of interpretation is essential. The system is meant to be individually-specific, tuned to its course context (e.g., the grading scale or the weights of each section as described in subsection 4.1.4). It is a tool that provides a greater wealth of information from which the instructor can make an informed decision, assuming the data from students is of good quality. Our philosophy mirrors NASA researcher Dr. Stephen Casner’s thoughts on autopilot: “We started with a creative, flexible, problem-solving human and a mostly dumb computer that is good at rote, repetitive tasks like monitoring. So we let the dumb computer fly and the novel-writing, scientific-theorizing, jet-flying humans sit in front of the computer like potted plants watching for blinking lights. It’s always been difficult to learn how to focus. It’s even harder now” [110]. Our desire is to let the instructor fly. Thus, our system assists by providing key information to the instructor for better decision-making.
The sentiment analysis algorithm itself is simple to use and can process reviews of a single student group’s work or multiple. Other optional parameters include sampling size, number of outliers to drop, lexicon to use (with or without stemming and weighting), and whether to provide a simple grade without supporting files (which include key word counts, quantitative stats, aspects, and a qualitative summary). SentiSoft is written in Ruby and leverages the Ruby port of UEALite Stemmer — a conservative stemmer for search and indexing [111] and the Ruby port of Lingua::EN::Tagger — an English Part-of-Speech Tagger Library [112].

Section 6.1 covers SentiSoft in detail, including the negation of sentiment, metrics collected, different grading methods with examples and discussion, and supporting documents generated. In section 6.2, we evaluate SentiSoft. Section 6.3 provides an experiment on review sampling to derive the minimum number of reviewers necessary for a reliable score from the sentiment analysis grading algorithm. In section 6.4, we furnish information on SentiSoft’s run time. Finally, section 6.5 summarizes the importance of our algorithm and process.

6.1 Algorithm

We processed single tokens, words or punctuation, and checked the lexicon in a specific order to determine their polarity as shown in Figure 6.1. Our approach captured sentiment at a unigram level and could be summarized per sentence, although we only tracked sentiment per review.

6.1.1 Negation

Sentiment negating words, or valence shifters, were perhaps the most complicated and interesting component of the sentiment grader and they occurred in just over 49% of our reviews. Words in our negate list were comprised of regular negation words (e.g., none), presuppositional words (e.g., barely), omission words (e.g., missing), and modal auxiliary
verbs (e.g., *should*). Negation was often associated with negative sentiment and not distributed equally with positive sentiment, a finding echoed in [113]. In our context, we found negating negative sentiment changed tone to neutral (e.g., *it wasn’t terrible* or *not wordy*) but negating positive sentiment changed tone to negative (e.g., *it wasn’t clear* or *nothing innovative*) [20], [41], [43], [114]. Our algorithm included a tunable *negate weight* parameter defaulted to 1 (i.e., we could reduce the polarity of negated words). The following actual
examples demonstrate the three ways our algorithm captured negated words [100] (see [27] for alternatives):

- Negate word to reset token
- Preceding negative qualifier (default distance of 4, see Figure 4.2)
- Trailing negative qualifier (default distance of 4, see Figure 4.2)

Certain words negated sentiment until their meaning was removed with a reset token (e.g., [115]):

“...could have given more practical examples to make it more clear as several readings were required to understand the topic and their idea flow.”

The positive words/phrases: practical examples, clear, understand the topic, and idea flow were all negated until the reset token, ‘.’, was encountered. Other reset tokens that conclude a negate word’s effect included: ‘.’, ‘;’, but, although, however, and nevertheless. Negate words included: no, not, can’t, nothing, hardly, barely, lack, more, suggest, miss, and few.

Secondly, polarity was negated through a preceding negative qualifier (similar to [100]). Simply put, a negative adjective closely prior to a positive word (“closely” is intentionally vague and was configurable in the algorithm):

“...which makes it difficult to understand...”

Finally, positive tone was negated through a trailing negative qualifier (as in [100]). This required observation into the future as the algorithm scanned the sliding window of text for a negative adjective which negated positive sentiment:

“...some insight was missing...”
It is important to note that the preceding/trailing negative qualifier was itself a negative sentiment word, thus “stacking” the effect of its negativity. This was intentional, as adjectives were often sentiment magnifiers [100], [116]. In contrast, negate words could either be negative or neutral, as seen in the overlap of the negate word lexicon in Figure 5.2.

6.1.2 Assigning a Grade

Figure 6.2 outlines our grading process, which extracted the analytical (i.e., review form questions) and subjective (i.e., review form comments) content into two files processed by different graders. The analytical grader simply matched student responses with their assigned values and aggregated the score. The process was straightforward and is detailed in subsection 4.1.4. We instead focus in this section on the subjective content grading process.

We utilized a variety of metrics and attributes to provide the maximum information to an instructor. Some are widely used in sentiment analysis (e.g., ‘tone’ and ‘purity’), while others were specific to our process (e.g., ‘dif’). The following metrics contributed to a sentiment score per review comment:

- weight: the sentiment of a lexicon-matched keyword: [0, 1]
- pos_keywords: the number of positive key words: 0 to inf
- neg_keywords: the number of negative key words: 0 to inf
- keywords: the total number of key words matched: 0 to inf
- tone: the sum of all weighted key words: (-inf, inf)
- info: the absolute value of all weighted key words: [0, inf)
- score: the sentiment score (F to A+): scaled from [-1, 1] to [0, 4.3]
The following attributes were defined or derived for validation and analysis per review comment:

- default: whether a comment could be scored: 0 or 1
- reliable: whether a comment could be considered reliable: 0 or 1
- dif: the difference between sentiment score and analytic section score: \([-4.3, 4.3]\)
- purity: tone over info, a measure of how consistent the sentiment was: \([-1, 1]\)
• positivity: the sum of positive sentiment: [0, inf)

• negativity: the sum of negative sentiment: (-inf, 0]

• negate_words: the number of negating words: 0 to inf

• words_per_sentence: the number of words per sentence: 0 to inf

• length: the number of words per review: 0 to inf

The following attributes were defined for validation and analysis for an aggregation of peer reviews:

• pos_lex_used: the percentage of words in the positive list utilized: [0, 1]

• neg_lex_used: the percentage of words in the negative list utilized: [0, 1]

Most importantly, our lexicon check for a single review comment provided tone and keywords. We assigned a fine-grained score to the text (Figure 6.3) by tone over keywords with some adjustments including scaling to the range desired by the instructor. Thus the highest scores were a result of having only highly weighted positive key words (i.e., high positive purity) and lowest scores were those having only highly-negative key words (i.e., high negative purity). We tested two options for aggregating comments by median and mean:

1. Weight reviews based on the information they contain (complex)

2. Weight all reviews equally (simple)

In the first grading scheme, we began by weighting our confidence in the comment. If the sentiment was negative but lacked many key words, we reduced the weight significantly to avoid penalizing harshly. If the review was positive with little information we weighted
slightly less to compensate. This was beneficial since a review with one key word would have the highest or lowest possible score depending on the polarity of the key word.

The second grading scheme in effect tested the principle of the wisdom of the crowd and allowed all reviews, even ones with a single key word, to contribute to a group’s score. This required a leap of faith — trusting that a few reviews with highly negative sentiment would not destroy a group’s grade. In either scheme, if there was not enough information to process the comment we simply incremented the number of default scores and the comment did not contribute to a group’s sentiment score.

Over the last five semesters, roughly 69% of our reviews met our basic threshold for scoring with weighted confidence. This number increased to \( \sim 90\%\) when we incentivized students to complete a quality review. Figure 6.4 is an actual example of the sentiment grader on a single group’s review (net positive – blue, net negative – red, negated – underlined), with the score and reliability generated from the complex scorer.

Finally, a third option was available for grading but left untested. If an instructor was willing to sacrifice the information contained in outliers, *dropping* could be used to remove the top and bottom \( n \) reviews from grading consideration. This would reduce the standard deviation and could possibly remove some extraneous reviews. According to our theoretical framework, however, we believe every review contains useful information and that outliers
should not be discarded. Thus even though the option was implemented, we did not test or consider it.

6.1.3 Grading Example

The following real review of one group’s essay submission with our sentiment grader and weighted lexicon is provided as an example to demonstrate the differences between the simple and complex scorers:

“This essay was a little bit short. It had a good spectrum of ideas but some details were missing. Also, more images could be added to make it more visually appealing. They did provide some great examples along the essay.”

The weighted key word output of the comment grader is as follows: “short (-0.3), missing (-0.5), (more) images (-0.4), (more) appealing (-0.6), examples (0.7)”. These values are summed to produce the following metrics and score:
\[
tone = -0.3 + -0.5 + -0.4 + -0.6 + 0.7 \\
\quad = -1.1 \\
info = |-0.3| + |-0.5| + |-0.4| + |-0.6| + |0.7| \\
\quad = 2.5 \\
keywords = 5 \\
weight = 0.25 \text{ (since } 2 \leq info \leq 3) \\
reliability = 1 \text{ (since } 4 \leq keywords) \\
purity = (tone/info) \\
\quad = (-1.1/2.5) \\
\quad = -0.44 \\
simple\_score = (tone/keywords) \\
\quad = (-1.1/5) \\
\quad = -0.22 \\
complex\_score = (tone/keywords) \times weight \\
\quad = (-1.1/5) \times 0.25 \\
\quad = -0.06 \\
\]

Mapping from [-1, 1] to the minimum and maximum values on the specific instructor’s range, [1.80, 4.30], produces a numerical score of 2.78 (B-) for the simple score and 3.01 (B) for the complex score.
6.1.4 Comparing Grading Schemes

When we simplified the grading scheme for two courses in Spring 2019 where students were incentivized to provide quality review comments, the means decreased on average, although the difference was minute: \( \sim 0.053 \) (1.2%) for Software Engineering (SE), an undergraduate course, and \( \sim 0.068 \) (1.6%) for Geometric Modeling (GM), a mixed undergraduate and graduate course. The medians were also affected, although less notably (0.68% and 0.15%, respectively). Figure 6.5 (average difference -0.068) and Figure 6.6 (average difference -0.053) show the difference in means using both scorers. One reason we saw some jumps between the grading schemes was that reviews with fewer key words (especially negative reviews) were previously weighted significantly lower (or even disregarded) so they would not sway the final score. A side effect of this simpler grading scheme was that our average percentage of reliable reviews went down along with our average positive sentiment and average length because we kept terser, more negative reviews.

Simplifying the grader also had the side effect of increasing the average standard deviation of SE by 2.00% to 0.435 and GM by 3.07% to 0.479. Although the average standard deviation was slightly high (above our target of 10% of the grade range, 0.430), the means were still accurate with a change of less than 2% (demonstrating the principle of the wisdom of the crowd). The fact that major algorithm changes (especially simplifications) only shifted grades within a fraction of a letter grade suggested that we had a robust sentiment analysis algorithm (partly due to the large number of peer reviews per work).

6.1.5 Discussion

We evaluated the Pearson’s Correlation Coefficient on each metric from SE and GM with the mean and median simplified sentiment scores and used a significance of correlation test (degrees of freedom \( df = 70 \) and critical value of \(+/- 0.380\)). Table 6.1 shows a selection
of correlation values \( r \) that are statistically significant at the \( \alpha = 0.001 \) level. Mean and median scores correlated very closely (0.935), suggesting that either may be utilized as the final score and that we have an approximately normal distribution. Purity was also highly correlated with mean which was expected in the simplified scoring algorithm.

To draw information from the simple sentiment scorer, we analyzed the correlation of mean with various metrics to infer the following:

### Table 6.1: Metric Correlations with Mean and Median

<table>
<thead>
<tr>
<th></th>
<th>FormScore</th>
<th>StdDev</th>
<th>NegSenti</th>
<th>Senti</th>
<th>Purity</th>
<th>NegKey</th>
<th>NegateKey</th>
<th>Adverbs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>0.750</td>
<td>-0.462</td>
<td>0.830</td>
<td>0.764</td>
<td>0.982</td>
<td>-0.814</td>
<td>-0.829</td>
<td>-0.373</td>
</tr>
<tr>
<td>Median</td>
<td>0.696</td>
<td>-0.281</td>
<td>0.820</td>
<td>0.696</td>
<td>0.913</td>
<td>-0.821</td>
<td>-0.813</td>
<td>-0.432</td>
</tr>
</tbody>
</table>
1. Negate Keywords had a greater negative correlation to the mean than Negative Words. This suggests that the role of negating words factors highly into grading, and situations in which they occur should be carefully handled.

2. Standard Deviation was negatively correlated with the mean. This suggests that most reviews are positive and that increased deviation comes from additional, negative reviews.

3. Percent Reliable had a very slight negative correlation with the mean (-0.127). This suggests that negative reviews might be more reliable than positive reviews.

4. Finally, a number of length and key word metrics were negatively correlated with the mean (significant at the $\alpha = 0.05$ critical value of $+/- 0.232$): Total Keywords (-0.249), Words (-0.284), Words/Sentence (-0.258), and especially Adverbs (-0.373).
This suggests that the more a student writes, the more faults they find in others’ works (even though just 15% of all reviews contained more negative sentiment than positive sentiment).

Taken together, this information indicated that many students wrote cursory, unreliable positive reviews, but a few diligent students wrote longer, more detailed, and more negative reviews. We find such negative reviews more helpful in an academic context and are interested in whether those reviewers grade more like an instructor or are higher-performing students. These reviews could be filtered to present to the instructor a more balanced and realistic perspective of each group’s work.

6.1.6 Supporting Documentation

SentiSoft produced much more than simply a numerical grade. There were a number of optional supporting documents generated, which would not be available in a peer review lacking sentiment analysis:

- A grade file that detailed how each comment was scored, including key words and weights

- A key word summary file that counted all instances of positive and negative key words mentioned

- A detail stats file that recorded twenty-seven different overall statistics for each review comment (i.e., the attributes from subsection 6.1.2)

- A summary stats file that provided the aggregate statistics from all reviews

- A mixed qualitative and quantitative half-page summary file (Figure 6.7) that included the mean comment score, a most-representative comment, and the top three aspects
(along with color-coded stemmed sentiment words and their context in sentences) that provided an overall summary of the comments

- A context file that listed all key words, color-coded according to polarity, with their context in the students’ comments

- A suggested aspects file that summarized potential new aspects for addition to the review form or lexicon

- An aspect summary file that outlined all aspects found, their number of instances, their total sentiment, and a list of their occurrences in context

6.2 Evaluation

Since sentiment analysis for a grade in the education domain is a new phenomenon and domain-specific labeled data sets were not available, we evaluated our subjective content grading process through qualitative analysis related to the instructor’s perspective (i.e., does the algorithm satisfy the one for whom it was developed). Quantitative analysis via a machine learning technique (neural network, NaïveBayes, tree, etc.) was not available for fine-grained scoring of unlabeled text.

There were a number of observations that hinted qualitatively at SentiSoft’s effectiveness (in addition to the information presented in section 5.4). Firstly, different grading algorithms produced scores that were almost identical (credit here belonged to the law of large numbers and a statistically sound procedure [117]). Using our simplified grader, the score was simply the scaled mean of weighted key words with weights determined by instructor heuristic. Lexicons are frequently built in this manner and instructor weights are assumed valid in academic testing [28], [27].
Actual review sentiment mean: 3.81/4.3 (A)
Most representative comment(s) scored at 3.8/4.3:

“The way they overcame the limitations using the piecewise polynomials is very much interesting. Technical details regarding B-Splines are explained in detail. Practical examples explaining computational algorithms helped me understand the topic more easily and effectively. The design curves and the step by step creation using the blender software is really great.”

Aspect **presentation** occurs 11 times with sentiment of approximately 1.0 (6.9/6.9):
- **engag:** “[Team’s] presentation was engaging as it started ”
- **engag:** “[Student’s] presentation was engaging and interesting . ”
- **informative:** “a very good informative presentation . ”
- **informative:** “the presentation very informative as the definitions ”
- **informative:** “presentation was very informative and has fine ”
- **clear:** “The presentation was clear and easy to ”

Aspect **explanation** occurs 5 times with sentiment of approximately 0.579 (1.1/1.9):
- **clear:** “good but a clear explanation is required ”
- **in-depth:** ”polynomial had an in-depth explanation which gave ”
- **proper:** “Also there is proper explanation and picture ”
- **slow:** “. Explanation was slow , banal and ”
- **slow:** “explanation was really slow . But , ”

Aspect **example** occurs 4 times with sentiment of approximately 1.0 (2.7/2.7):
- **crisp:** “opinion , including crisp examples , nice ”
- **practical:** “shapes as a practical example and theoretical ”
- **interest:** “design examples were interesting , allowing for ”
- **relevant:** “examples are very relevant and intuitive . ”

**Figure 6.7:** Team Review Summary Example

Secondly, we noted an infrequency of instructor adjustment to the algorithm score. There were three options to handle differences between the algorithm (crowd) and instructor score: 1) accept the wisdom of the crowd after we had established the algorithm and the minimum number of reviewers, 2) use the crowd score to adjust the grade of the instructor, or 3) adjust the crowd score. In five semesters of use (nine courses, over 300 student submissions), the instructor only changed the algorithm’s score *once*. In that situation, the score was adjusted up to compensate for a number of students reporting a failing score for “reading
off the slides”. The issue was further compounded by the rather coarse-grained overall score selection (a scale of 4 instead of the current scale of 8) from a previous review form iteration. Increasing the granularity of the overall score softened the otherwise large variation.

Thirdly, the sentiment score roughly tracked the review form score. In four different courses, the review form score was slightly higher on average, ranging from 1.9% to 4.7% of the total score, than sentiment mean (Table 5.3). We expected some deviation between the scores — a phenomenon also observed in the product reviews domain [18] — as students were willing to write honestly, even if they gave their peers a good grade via the analytical section.

Finally, reviews carried enough information to be counted. In the SE and GM analysis (subsection 6.1.5), we average 4.24 positive key words, 1.48 negative key words, and 1.13 negate words per review. Percent default scores ranged from 0% to 13% (3.6% on average for GM and 2.1% for SE) with 63% of reviews marked as reliable using the simplified scorer. Visual evidence of domain reliability can be found in Figure 5.3. When we combine the number of reviews marked reliable with our average of 29.44 reviews over the past five semesters, this translates to approximately 18.55 reliable reviews per work which is comfortably past the threshold in section 6.3 for a low average error and standard deviation.

### 6.3 Sampling

To the best of our knowledge, no research has quantitatively measured the ideal number of reviewers in peer review. Perhaps this is because peer review in the conventional form places an increasing burden on students per additional reviewer, thus most works have 2-4 reviewers per project or essay. Our method of reviewing to learn (chapter 3) does not have this fundamental limitation and we gathered an excess of reviews, up to 35 or 40 per work. This allowed us to experimentally determine via sampling if a fewer number of reviewers would suffice.
Table 6.2: Graduate Review Sampling

<table>
<thead>
<tr>
<th>Reviews</th>
<th>MAE</th>
<th>Stddev</th>
<th>MdAE</th>
<th>Positive Direction (%)</th>
<th>MAE</th>
<th>Stddev</th>
<th>MdAE</th>
<th>Positive Direction (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>9.766</td>
<td>10.863</td>
<td>9.895</td>
<td>0.486</td>
<td>10.512</td>
<td>11.889</td>
<td>10.772</td>
<td>0.556</td>
</tr>
<tr>
<td>3</td>
<td>7.926</td>
<td>8.235</td>
<td>8.535</td>
<td>0.457</td>
<td>8.737</td>
<td>8.870</td>
<td>9.337</td>
<td>0.528</td>
</tr>
<tr>
<td>4</td>
<td>6.840</td>
<td>7.047</td>
<td>7.088</td>
<td>0.429</td>
<td>7.341</td>
<td>7.111</td>
<td>8.107</td>
<td>0.667</td>
</tr>
<tr>
<td>5</td>
<td>5.926</td>
<td>5.921</td>
<td>6.458</td>
<td>0.429</td>
<td>6.444</td>
<td>6.537</td>
<td>7.130</td>
<td>0.500</td>
</tr>
<tr>
<td>10</td>
<td>3.721</td>
<td>3.517</td>
<td>4.106</td>
<td>0.600</td>
<td>4.014</td>
<td>3.702</td>
<td>4.663</td>
<td>0.528</td>
</tr>
<tr>
<td>15</td>
<td>2.553</td>
<td>2.479</td>
<td>2.873</td>
<td>0.457</td>
<td>2.753</td>
<td>2.484</td>
<td>3.191</td>
<td>0.528</td>
</tr>
<tr>
<td>20</td>
<td>1.848</td>
<td>1.771</td>
<td>2.149</td>
<td>0.457</td>
<td>1.778</td>
<td>1.446</td>
<td>2.221</td>
<td>0.472</td>
</tr>
</tbody>
</table>

Table 6.3: Undergraduate Review Sampling

<table>
<thead>
<tr>
<th>Reviews</th>
<th>MAE</th>
<th>Stddev</th>
<th>MdAE</th>
<th>Positive Direction (%)</th>
<th>MAE</th>
<th>Stddev</th>
<th>MdAE</th>
<th>Positive Direction (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>11.418</td>
<td>12.171</td>
<td>11.610</td>
<td>0.528</td>
<td>9.259</td>
<td>9.358</td>
<td>9.371</td>
<td>0.667</td>
</tr>
<tr>
<td>3</td>
<td>9.109</td>
<td>8.514</td>
<td>10.793</td>
<td>0.389</td>
<td>7.492</td>
<td>6.827</td>
<td>8.410</td>
<td>0.528</td>
</tr>
<tr>
<td>4</td>
<td>7.451</td>
<td>6.731</td>
<td>8.560</td>
<td>0.444</td>
<td>6.397</td>
<td>5.792</td>
<td>7.170</td>
<td>0.500</td>
</tr>
<tr>
<td>5</td>
<td>7.081</td>
<td>5.663</td>
<td>8.444</td>
<td>0.417</td>
<td>5.540</td>
<td>4.767</td>
<td>6.691</td>
<td>0.472</td>
</tr>
<tr>
<td>10</td>
<td>4.053</td>
<td>2.974</td>
<td>5.284</td>
<td>0.528</td>
<td>3.298</td>
<td>2.867</td>
<td>4.196</td>
<td>0.583</td>
</tr>
<tr>
<td>15</td>
<td>2.966</td>
<td>2.121</td>
<td>4.004</td>
<td>0.444</td>
<td>2.431</td>
<td>1.928</td>
<td>3.217</td>
<td>0.472</td>
</tr>
<tr>
<td>20</td>
<td>1.818</td>
<td>1.229</td>
<td>2.744</td>
<td>0.639</td>
<td>1.680</td>
<td>1.183</td>
<td>2.087</td>
<td>0.472</td>
</tr>
</tbody>
</table>

Table 6.2 and Table 6.3 show, on a 100-point scale, the mean absolute error (sampled mean minus actual mean) for different sampling levels in both undergraduate and mixed graduate and undergraduate courses. Sampling results were averaged over 20 runs for an entire semester each (i.e., 36 works). Positive Direction refers to the percentage of actual sampled means that were greater than the actual means and shows the model bias.

Rather than dogmatically prescribing a global threshold of reviewers for all situations, the goal of this section is two-fold: 1) to observe quantitatively the optimal number of peer reviewers using our methods for scoring confidence and 2) to suggest that using the conventional 2-4 peer reviewers may limit accuracy [72]. Figure 6.8 shows the average error and standard deviation for different sampling levels over all four courses (two semesters, 144 student works). For an average error consistently less than 5%, 15 reviews were required. For an excellent average error (and standard deviation) less than 3%, at least 20 reviews were required.
6.4 Run Time

Run time is an important issue when considering scalability. Adding students to our courses did not increase the grading burden in the typical fashion since we used peer review and the work was distributed, but it did add to the algorithmic cost. This section covers the run time of each component, from real workloads, on the following workstation typical of an instructor's personal computer:

- Windows 8.1 operating system
- Intel Core i7-4722HQ 2.40GHz CPU with 4 cores
- 8GB of SODIMM 1333 MHz memory
Wireless router speed averaging 80Mbps download and 10Mbps upload

6.4.1 Email Scraping

By far, the most compute-intensive aspect of the grading process was using the Google API to scrape HTML reviews emailed from the website form. The reviews were gathered and parsed into three different files: a comments file, a grade file for the analytical grader/score, and a participation file that linked each student’s comment to their student I.D.

Writing the files was fast, but this process was highly dependent on the internet connection speed and search period (i.e., the number of emails to sort through). In our environment, for a 3.5 week submission period gathering 20-45 essay or term project reviews this took approximately 6-7 minutes per team assignment. For a 2 week submission period gathering 30-45 presentation reviews this took approximately 1.5 minutes per team assignment.

6.4.2 Subjective and Analytical Graders

We used the Ruby Benchmark gem to time our algorithms’ average elapsed real time over 25 iterations for both our sentiment and analytical graders. Both algorithms utilized hash tables (for the lexicon lookup in the subjective grader and the answer weight lookup in the analytical grader) and other standard performance-maintaining techniques.

Figure 6.9 shows the average run times for both algorithms, process reviews and analytical grader, for a variable number of reviews on a single team’s work. When processing review comments, there is the option to not write supporting documentation (i.e., grade only). Ultimately, the average time to process all reviews was fairly linear. Process reviews increased by 0.40 seconds per ten reviews, process reviews with grade only increased by 0.24 seconds per ten reviews, and the analytical grader increased by 0.26 seconds per ten reviews.

Figure 6.10 demonstrates the rate at which the algorithms increase in elapsed real time when bulk grading multiple teams’ works. In this case each team had 46 reviews, our
current worst-case scenario. Up to 27 projects, we noticed a fairly consistent linear increase. Overall, process reviews increased by 5.96 seconds per three projects, process reviews with grade only increased by 3.59 seconds per three projects, and the analytical grader increased by 1.78 seconds per three projects. Since we typically had a maximum of two courses with 12 teams per course, our maximum number of projects to grade at any one time was 24 (when grading essays or term projects, which were due at the same time in both courses).
6.4.3 Increasing Performance

Both algorithms — process reviews and the analytical grader — called one another. The analytical grader called process reviews to gather the sentiment score if it was weighted into the final score. Process reviews called the analytical grader to obtain the numeric grade for comparison in the supporting documentation. Note that the run time values could be improved in both Figure 6.9 and Figure 6.10 if these algorithms were modified to not call one another (i.e., do not consider the comment score for the final, weighted grade and/or do not generate supporting documentation). In our case, we preferred the increase in information to reducing run time further.
6.5 Conclusion

In this chapter, we presented a detailed overview of our application for peer review sentiment analysis of subjective content. We advocate our process of using, rather than ignoring, sentiment in peer review text — an untapped area with great potential for increasing information and reliability in assessment. Through sampling experiments we infer that 2-4 peer reviews are inadequate for grading reliability. In contrast, we have shown SentiSoft, with 15-20 peer reviewers, is reliable across many semesters — even using different grading schemes — with excellent run time performance. Finally, our algorithm is highly configurable and easily adaptable to a local context (see chapter 7 for an example). It is our desire that other educators will test SentiSoft, employ it in their classrooms, and improve upon it for their own benefit.
Chapter 7: An Application of SentiSoft

“Generalization is necessary to the advancement of knowledge; but particularity is indispensable to the creations of the imagination.” – Thomas B. Macaulay

We have shown that our peer review process works well in the context of our specific courses (software testing, software engineering, computer graphics, and geometric modeling). However, the relevance of SentiSoft to other course contexts would greatly improve its usefulness and reach, since not every instructor may be eager or able to reformat their courses and utilize our educational principles and methodologies from chapter 3. The following is a description of peer review in a data visualization course taught at the University of South Florida from 2017-2019 that utilized SentiSoft to answer research questions. It includes the application of 1) our aspect extractor and 2) our sentiment analysis algorithm to derive insights from a course with a different instructor, domain, and peer review process. Thus, this chapter is experimental evidence for the wider applicability of our algorithm in the context of peer review.

Section 7.1 introduces the course domain, data visualization, and notes the benefits from and absence of peer review research in the visualization community. The course was not designed according to the educational principles in chapter 3, the rubric (used in lieu of “review form” to match the visualization domain terminology) was not created as described in chapter 4, and the lexicon was not built from the peer review text as in chapter 5. Therefore, the course format (section 7.2) and peer review process (section 7.3) are covered in detail. Additionally, sentiment analysis was not used to produce a fine-grained sentiment

\[5\text{This chapter was published in [12]. Permission is included in Appendix A.}\]
score from crowd-sourced peer review text as detailed in chapter 6. Instead, our algorithm was used as post-processing of the review comments to draw information from students who took the course (section 7.4). Finally, section 7.5 summarizes the benefits of peer review in the data visualization domain and the value SentiSoft provides.

7.1 Motivation

Rushmeier et al. defined visualization education as a work in progress [118]. Nevertheless, the subject can be broadly split into two categories. The first is the proper construction of visualizations — using the right algorithms and visualization principles, which tends to be the primary focus of visualization courses in which student comprehension of concepts, techniques, and algorithms can be objectively measured. The second category is focused on the subjective evaluation of the quality and accuracy of visualizations. Subjective evaluation is not only important for the instructor’s assessment of students but for students to develop the ability to evaluate others’ visualizations critically. These skills are commonly taught through informal methods, such as group or whole-class discussions that can leave students’ skills underdeveloped [119].

Furthermore, with Gen Z learners (individuals born between the mid-1990s and mid-2000s), educational preferences have shifted significantly. They prefer instant feedback, are increasingly collaborative, and are active-learners who prefer project-based coursework [120]. Meanwhile, with a large number of students enrolled in visualization courses, it is difficult to provide students the timely, subjective feedback they need to improve the quality of their work [121].

Peer review is a highly-engaging feedback mechanism [122], [123], often used in liberal arts courses [124], [125], [126], [127], human-computer interaction (HCI) courses [128], [129], [130], code review [131], and scholarly publication [132], and it is ideal for addressing these challenges. Instead of relying solely on instructors for feedback, peers collaborate to provide
diverse multi-sourced feedback to one another with a relatively quick turnaround. In addition, the evaluation process itself gives students an opportunity to reinforce recently-learned course concepts by critically evaluating others’ work. Finally, for instructors, it is a scalable approach. Since students provide feedback to one another, adding students to a course only adds to nominal administrative efforts.

Despite these well-known advantages of peer review, we found little evidence of peer review in visualization courses. In a survey of one hundred information visualization faculty, we found eighteen publicly available course syllabi, only one of which mentioned peer review. In order to address this gap, we discuss the construction and evaluation of a peer review-oriented computer science visualization course, to encourage the visualization community to initiate discussions around and to eventually adopt this pedagogical methodology into their classrooms.

7.2 Course Overview

With the extensive usage of peer review in other disciplines, we endeavored to build a visualization course with peer review as a core component. Our course, titled “Data Visualization”, was taught Spring semester of 2017, 2018, and 2019 and was a co-listed undergraduate and graduate course. The course was hosted in the Computer Science department, within the College of Engineering at the University of South Florida, and the educational emphasis of the course was primarily good visualization practice, though a strong emphasis was placed on software design as well.

7.2.1 Course Content

The content of the course was primarily taught following Munzner’s *Visualization Analysis and Design* [133] and the Nested Model [134] with additional outside visualization content (e.g., Vis Lies and New York Times Graphics) and research papers added throughout the
The presentation methods consisted of lecturing, research paper presentations by students, and discussions (e.g., small group and whole class critiques).

The course learning objectives were that students would demonstrate the ability to:

[L1] Build effective visualizations by evaluating a provided data and user requirements and programming an interface to match those requirements.

[L2] Associate visualizations with the foundational components, e.g., data abstractions and visual encodings, that go into their construction.

[L3] Critique the effectiveness of interactive visualizations with respect to task selection, visual encoding choices, and interaction design and implementation.

To satisfy [L1] and [L2], the course consisted of eight projects, one in Tableau and seven in Processing, totaling 50% of their final grade. Projects were designed so that students could reuse code and implement peer feedback and were typically due every 10-14 days. Upon the completion of each project, students were asked to provide reviews (satisfying [L3]) of three randomly selected peers’ work using a provided rubric within 5-7 days (see section 7.3). The peer feedback served as the primary form of qualitative feedback students received. In addition, the Spring 2018 version of the class included self-review after the completion of peer review. Both peer review and self-review were for small amounts of (spot checked) completion-based credit (approximately 10% of the final course grade). The instructor and teaching assistants still completed project grading. Grades were assigned based primarily upon objective requirements, as well as some subjective judgment.

7.2.2 Evaluation Methodology

There are significant subjective measurements necessary to evaluate the effectiveness of a visualization course [135]. Visualization education research also lacks a standard rubric
for measuring engagement. In a review of five major visualization venues (InfoVis, SciVis, VAST, EuroVis, and Pacific Vis), we found no form of empirical evaluation of students’ work or engagement in the classroom. Thus, we have conducted a qualitative evaluation using questionnaires to measure perceived improvement and engagement. We supplement and reinforce the analysis by combining the questionnaire data with both qualitative analysis using representative examples of student work and quantitative analysis through natural language processing.

A useful reference for our non-experimental evaluation methodology can be found in the field of writing composition. For example, Mulder et al. used similar qualitative and quantitative methodologies, based on content analysis from peer review comments and student questionnaires [136]. Furthermore, analyzing the number and variety of words used in written comments has been used for measuring both learning outcomes [137] and student engagement [138].

7.2.3 Data Collection

The student-produced data was collected from a number of categories. Student visualization project submissions were gathered from eight projects each year (2017-2019). We manually reviewed projects to find those which, in combination with peer comments, exemplify the value of peer review.

Student peer review comments were collected on eight projects from 2017 and seven projects from 2018 and 2019. Student self-review comments were collected on seven projects from 2018 only (self-review was dropped in 2019 due to concerns about workload). We analyzed them with SentiSoft which produced numerical feedback including overall sentiment of the text, counts of parts of speech (i.e., noun, adjective, adverb), the average length of comments, etc. The algorithm includes an aspect extractor (see section 4.2) that was developed for peer review in engineering courses, but not explicitly tuned for visualization.
Figure 7.1: Histograms of Questionnaire Respondent Demographics. (a) students per year (n=139), (b) gender (n=129), (c) undergraduate vs. graduate (n=111), (d) anticipated course grade (n=137), (e) times using peer review in a CS course (n=138). Color-coding: darker colors indicate earlier years.

Each year (2017-2019), students were asked to complete an optional 20-question post-course questionnaire, given after their final exam and before receiving their final grade. Students were offered a small amount of extra credit to anonymously answer questions ranging from demographic information and anticipated final grade (see Figure 7.1) to helpfulness of review comments and open-ended suggestions for improving the peer review process. Numerical answers were all placed on a 5-point Likert scale. The overall participation rate was 98%. We performed quantitative analysis on numerical answers and manually reviewed written feedback for qualitative results.

7.3 Project and Peer Review Design

When designing projects, our interest was to see students gain visualization skills (i.e., [L2]) by demonstrating proficiency in using software engineering problem-solving techniques (i.e., [L1]) [139]. The peer reviews served as an integral part of the future projects by allowing students to use feedback to refine the work they submitted (i.e., [L1]). Therefore, the projects
were set up to maximally build upon and reuse components from previous projects, while still challenging students with new project requirements. Visual examples of the progression of projects (from simple bar and line charts to complex dashboards) can be found in Figure 7.9 and Figure 7.10.

One point of variation between the semesters was the platform used for code submission and subsequent peer review. For project submissions, we experimented with zip files via Canvas assignments (2017), Bitbucket (2018), and GitHub Classroom (2019). Peer review in 2017 was handled via Canvas’ built-in peer review platform, which provides a liberal arts style of peer review, where a document is displayed, and questions appear alongside. In our case, students had to download and run code on their local machine. Once feedback was submitted, it would immediately become available to the recipient. In 2018, peer reviews were assigned and submitted via a Canvas quiz. At the end of the peer review period for a project, the feedback was returned by e-mail using custom scripts. In 2019, Google Forms was used to capture feedback and delivered via a custom webpage at the end of the peer review period.

Figure 7.2 shows the post-course survey response to the question of whether the interface “worked well”. These are independent samples, with no common baseline, making direct comparison impossible. However, the students using the Canvas peer review system had the most favorable view of that platform (see 7.2(a)), followed by the Google Forms group (see 7.2(c)), then the Canvas Quiz group (see 7.2(b)). In the free-response section of the survey, several students in the Google Forms group stated that better integration with Canvas would have improved the experience.

7.3.1 Peer Review Rubric

The rubric was built by carefully reviewing the course content and extracting key concepts necessary for demonstrating proficiency in [L2] and [L3] ([140], [141]). The basic structure
of the rubric divided topics into five major assessment categories, with each category having three sub-assessments affixed to a 5-point scale. Each sub-assessment contained a comment box for details on the scoring.

The five major assessment categories are algorithm design, interaction design, visual design, design consideration, and visualization narrative. The algorithm design category examines algorithm selection and implementation. Interaction design relates to user interaction with the visualization. Visual design checks the technical aspects of data placement in the visualization (e.g., visual encoding channels, their expressiveness, and their effectiveness). Design consideration focuses on the composition and aesthetic aspects of the visualization, such as embellishments. The final category, visualization narrative, is used in projects where the story surrounding the visualization is as important as the visualization itself.

In the original design of the rubric, we intended a certain level of customization to be applied based upon the content of an assignment or course. For each project, we extracted the relevant components from the full template. Project 1, for example, included a narrative component, while no other project included such a requirement. Projects 4-8 had interaction components, while Projects 1-3 did not. An example can be seen in Figure 7.3 (right) for the assessment received from four peers to the submission in Figure 7.3 (left). The example
shows three main categories — visual design, design consideration, and interaction — along with sixteen sub-assessment questions for Project 6.

**Figure 7.3:** Dashboard and Qualitative Feedback. Example submission for Project 6 “Building a Dashboard” with a visualization of the qualitative feedback received from four peers.

The rubric covers all of the key concepts instructors intended for students to master. On the post-course questionnaire, we asked the students three questions related to their opinions of the rubric (see Figure 7.4). Their opinions indicated that, while the rubric questions were useful (see 7.4(a)), fewer (see 7.4(b)) with the same level of detail (see 7.4(c)) would be preferred. Methodologies for reducing the number of questions can include combining similar sub-assessments over the course of the semester.

### 7.4 Peer Review and Student Learning Outcomes

While assessment is the heart of formal higher education and a core component of effective learning [142], we did not prioritize evaluating the correctness of a visualization or how it
corresponded to an instructor’s grade. Rather, we were interested in evaluating the influence peer review had on students producing the visualizations. Specifically, we limited our analysis to three questions:

1. Does peer review reinforce course content?
2. Do students engage in and enjoy the peer review process?
3. What aspect of peer review is most beneficial to students?

7.4.1 Does Peer Review Reinforce Course Content?

We primarily utilized the student peer review comments to determine whether peer review reinforces course content. If students mentioned key concepts learned in the course in written responses, we interpreted it to mean that they took the opportunity to identify course content in context (i.e., [L2]).

The open-ended review comments (peer=3104, self=339, total=3443) were gathered from multiple comment sections on each rubric. Each comment section was concatenated into a single string and analyzed by the process described in subsection 7.2.3.

We first compared whether students were using terminology from the rubric or whether they were commenting on other things. Included in the top twenty aspects were the following
words: visualization, legend, color, ink, data, type (of data), graph, information, ratio (of data), use (of color, encoding), scale, amount (of ink, data), interaction, chart, lie, and density. These words are all found on the rubric, so students were likely parroting the terminology. Nevertheless, we posit that this forces the students to identify course content in the context of their peers’ work. Figure 7.5 provides context for the utilization by displaying aspects in the center column with connected positive words (left column) and negative words (right column). The height of the bars indicates the number of occurrences. Color saturation indicates how positive or negative the sentiment is.

Figure 7.5: Visualization of Peer Review Sentiment

Although we cannot directly measure whether students understand the concepts better through peer review, we can be confident that this approach provides repeated exposure to
key concepts (giving and receiving on-topic feedback), and students are, at the very least, repeating terminology.

Finally, comments from the post-course questionnaire like the following seem to confirm that students note course content in each others’ visualizations:

“I’d definitely recommend keeping it . . . It also helps those struggling with concepts to see how others did it, to do it better on future assignments.”

“Peer review helps in understanding data visualization principles . . . This helps a lot in doing future assignments and understanding my mistakes.”

7.4.2 Do Students Engage In and Enjoy Peer Review?

To quantitatively evaluate student engagement, we analyze the number and variety of words written [138]. We specifically tag nouns (aspects), adjectives (aspect modifiers), and adverbs (sentiment enhancers) from the peer review comments. The relevant summary statistics are shown in Table 7.1.

We noticed that undergraduates wrote more than graduate students, with more words per sentence and a greater variety of tagged parts of speech (especially adverbs). Interestingly, the ratio of negative to total keywords was similar for undergraduates (38%) and graduates (36%). This is an important measure of engagement because critically evaluating a visualization requires more investment than just a cursory review — it requires applying learned concepts to explain why something is wrong (e.g., analyzing for “lie factor”). Considering the length, variety of parts of speech, and the ratio of negative keywords may
indicate that undergraduate students are slightly more invested in the peer review process than their graduate peers. Many post-course questionnaire comments reflected a general sense of increased engagement and motivation:

“Process is interactive and healthy.”

“I enjoyed it! Saw some really good work by my peers that motivated me in the final projects.”

“Definitely worth the time and energy, learned a lot by examining code, which helped to see their thought process.”

To discover whether students enjoyed the peer review process, we asked two questions on the post-course questionnaires (see Figure 7.6): 1) if students believe they learned more because of the peer review process (perceived improvement); and 2) if students recommend continuing peer review. 82% of respondents reported learning at least somewhat more (score of three or more; mean = 3.6).

Additionally, 75% recommended continuing the process (score of four or more; mean = 4.1) with almost half of students strongly recommending it, despite the fact that reviews were
more work, taking between 10 and 15 minutes to fill out (see 7.6(c)). Finally, some students enjoyed the process so much that they recommended increasing the stakes of peer review in the course:

“I liked being able to view what others did because it helped me see ways I can improve my work. I also think the peer reviews should be worth more points.”

“I think you should showcase the best and worst visualizations in class. Let the students praise/rip apart them as an exercise. It’s also more motivation to do well.”

7.4.3 What Aspect Is Most Beneficial to Students?

To determine which aspect of peer review is most beneficial to students, we asked three questions about the helpfulness of content: receiving feedback (7.7(a)), providing feedback to others (7.7(b)), and participating in self-review (7.7(c)). Interestingly, in order from most helpful to least, most students found 1) reviewing others’ work (mean = 3.9); then 2) self-review (mean = 3.4); and finally 3) feedback received (mean = 3.2) helpful. Thus, it appears that students perceive the maximum benefit from their ability to review others’ work ([L2] and [L3]), not by receiving feedback on their own, which is consistent with Garousi’s findings [143]. Students leaned towards preferring textual (as opposed to numeric) responses (see 7.8(a)), but a surprisingly low rating was received for the quality of peer reviews (mean = 3.2) (7.8(b)) which may explain why nearly 25% of students never or rarely looked at their feedback (7.8(c)).

Finally, many students mentioned explicitly in the course questionnaire how helpful it was to view others’ code and visualizations:

“Looking at other people’s code helped. Grad students = bad code.”
“I think it’s a good idea; it helped me by giving examples of what not to do mostly.”

“I am very impressed with this process. It is very helpful to students.”

“The way it’s done gives you a chance to learn from many different peers as well as help teach many.”
7.4.4 Peer Review Examples

Although we did not provide a control group for peer review, the following examples of particular students’ progress throughout the semester ([L1]) reflect a combination of instruction and the effectiveness of the peer review process.

In the first example, we highlight differences between Projects 2, 4, and 6 to demonstrate the effect of a student receiving and implementing peer feedback. 7.9(a) received the following comments: “the text size used for labels could have been more larger [sic] for clear vision. [...] in bar and line chart, the line points could have been brought into center” and “there wasn’t any color present until the combo chart separated the two chart types with a red color. However, the data seemed bland when presented in black and white.” In response, the student changed the color of the bar chart, moved the line chart points to the center of the label (actually displaying the points themselves), and made the axis titles slightly larger, as shown in 7.9(b). 7.9(b) received the following comments: “somehow line graphs and scatter plots are shifted” and “I believe using at least one other color would make the
charts even more appealing”. In response, the student added another color to 7.9(c) and shifted the appropriate graphs to not overlap. Thus, in addition to benefiting from reviewing others’ work, the student appeared to consider and implement much of the feedback they received.

Another interesting observation from reviewing feedback was that students tend to comment on others’ work in avenues that they have already implemented in their visualizations (showing progress in [L2] and [L3]). Figure 7.10 shows student projects and the feedback the student gave to their peers. For example, in Project 2, the student mentions axis ticks and labels — something they carefully implemented in their bar and line charts. In the Project 3 feedback, the student points out the correct use of colors and directly references materials learned in class. The student references a specific programming technique for avoiding clutter in a PCP in their Project 6 feedback, and finally, a tip to delineate charts on the dashboard for Project 7 feedback. In each situation, the student offers advice that corresponds to a technique they correctly implemented, which corroborates our findings in subsection 7.4.1 that peer review reinforces course content by allowing students to communicate recently learned and applied concepts during the peer review process (an opportunity they might not otherwise have).

7.4.5 Instructor Perspective: Observed Student Benefits

The previous examples have shown, from student data, how peer review benefits visualization students. We have observed several additional student benefits that we informally evaluate.

*Formalizing Existing Collaborations.* Many, but not all, students will discuss projects with one another and seek feedback without instructor intervention. However, peer review using a rubric forces (in a positive way) interaction for some students who would not otherwise interact with and offer constructive criticism to others. To some, it offers a more comfortable
environment than discussion in class. In all cases, it provides structure to the feedback. Requiring feedback in this way is important to the student (to practice critical evaluation skills), as well as to the visualization (a greater variety of perspectives contribute to increased effectiveness).

**Increasing Ownership.** Multiple students commented on the value of seeing each others’ work. There are two effects that we observed: 1) it enables seeing how peers have solved a problem in order to redesign your visualizations; and 2) there is pressure to perform better when you know your peers will see your work. From our conversations with students, we felt that this gave the students more ownership in the projects, many discussing with pride the final product they developed.

**Critical Evaluation Skills.** Critical evaluation of others’ work is a mandatory skill for academia, for assessing research quality, and in industry, for code and design reviews. Many engineering programs are knowledge and skills-based (ABET accreditation criteria reinforce this) while teaching critical evaluation informally. Peer review can help fill this gap in developing skills.
7.5 Conclusion

This chapter presented a unique application and evaluation of peer review in a different domain: the data visualization classroom. The approach highlighted two significant benefits. First, it supplied a framework for engaging students through critical evaluation of visualizations. Second, it was a mechanism for providing students with diverse and timely feedback on their work. It is interesting to note that, while students do not always know or appreciate best practices for their education, they do enjoy peer review (Figure 7.6) and believe that it is helpful for their learning (Figure 7.7).

SentiSoft was an instrumental tool for gathering quality information from the peer review comments, after the course completion, to provide insights into whether course content is reinforced through peer review (Figure 7.5) and whether students are engaged (Table 7.1). This emphasizes the flexibility of our algorithm, which can be used for a variety of objectives beyond its original purpose.
Chapter 8: Conclusion

In this dissertation, we have covered the introduction of a novel application of sentiment analysis in the classroom. Unlike other recent approaches that focus on student attrition, the mood of students, negative students, teacher strengths or weaknesses, or student perception of industry experience — all very helpful indicators for the improvement of students’ academic experience — we focus on utilizing sentiment analysis in a fundamentally different way. We use a peer-reviewing crowd to intelligently assess a group’s work (presentation, essay, or term project) and then use natural language processing techniques to quantitatively score peer review text. This assists with the scalability of course assignments and provides extra analytics for grading. While our algorithm is helpful primarily to the instructor, it indirectly improves the overall course experience for students when used with our teaching methodology (i.e., students learn by critically reviewing peers, receive diverse and timely feedback, and develop soft skills through the practice of peer instruction).

In chapter 3, the current state of assessment scalability in courses with open-ended projects, along with the inadequacy of Automated Essay Scorers, was presented. In contrast, we put forth a learning environment with associated tools to effectively wield students as unused capacity, to pursue risk with graceful failure, to present research, and to critically evaluate their peers’ work. They do so using a review form designed to sample their knowledge effectively.

We then detailed exactly how the review form was created and revised in chapter 4. We approached the iterative process in a data-driven way by first understanding what students were trying to communicate. This information was gathered from the open-ended comment
section and analyzed/interpreted initially by hand, but later semi-automated by our aspect extractor. The aspect extractor compiled a list of frequent key words associated with positive or negative sentiment. In this way, potential topics to add to future iterations of the review form were found, as were ways to clarify current questions to clear up any student confusion. Although our aspect extractor analyzes text differently than others (i.e., it first looks for sentiment-bearing words, then for potential noun aspects in the vicinity instead of vice versa), it performs well on an external gold standard data set with a high precision and moderate recall, even finding important aspects not previously discovered.

Essential to both the aspect extractor and the sentiment analysis algorithm was HeLPS, our key word lexicon presented in chapter 5. Without focused, domain-specific key words, precision would be low and accuracy would suffer. The key words were gathered from the student review comment data by leveraging human intelligence to infer contextual polarity (weight and direction of sentiment). We tracked the specific key words mentioned in reviews per student project and per semester to determine what students emphasize. Finally, we compared HeLPS to six other publicly-available weighted and unweighted lexicons to determine accuracy and usefulness. Unsurprisingly, HeLPS was the clear winner, both quantitatively and qualitatively. This highlights the importance of domain-specific tools, as many others have emphasized (chapter 2).

Leveraging the lexicon, our sentiment analysis algorithm, SentiSoft, was introduced in chapter 6. We covered various steps including the negation of sentiment, metrics captured, and two different grading strategies. We also noted the supporting documentation which accompanied our peer review analysis to provide an instructor with much more information than would be available without a lexicon-based approach. Finally, we evaluated our algorithm, performed a sampling experiment to determine the fewest reviewers necessary for accuracy, and documented the run time.
Chapter 7 was an important experimental indicator of our algorithm’s applicability to other course contexts. SentiSoft was utilized in another instructor’s data visualization course that included peer review to deliver valuable insights into 1) student engagement and 2) student identification of key course concepts. Without our algorithm, such analytics derived from student text would be quite difficult. This references the utility of our work even if our educational methodology is not followed to the letter.

Throughout our work, we have emphasized the importance of following the process, rather than copying the result. For example, our review form was built from the student peer review comments particular to our courses, and should not be assumed valid in another course. An investment in mining valuable information from peer review text will yield a precise system.

As we look to the future, we note a strong need for good quality, labeled datasets for text in the student peer review domain. With gold-standard (ground truth) data for academic peer reviews, sentiment algorithms like ours can be validated more easily. The true value of sentiment in a review comment, or perhaps an average of true values from independent observers, can be reached through labeling by a small crowd (e.g., Amazon Mechanical Turk).

Within our domain, we are interested in improving our process in a number of ways. Firstly, collecting reviews via the Google API is slow and should be streamlined. Secondly, we would like to incorporate a set of intensifiers to the lexicon for accuracy and to balance the effect of negation. We would also like to expand our lexicon to include phrases (bigrams, trigrams, etc.) rather than single key words (unigrams). It would be interesting to compare the polarity of words in this augmented lexicon to those in other lexicons. Perhaps the inaccuracy of larger lexicons comes from including irrelevant words rather than the incorrect polarity of relevant words. If so, a strategy to select a limited number of related words from an existing lexicon could be used to quickly generate a domain-specific lexicon in a novel area. Finally, we are interested in a hybrid approach to text scoring mixing two or more lexicons, especially those with different strengths (e.g., SlangSD captures negative comments...
well). The results on a given phrase or sentence can then be voted on by the lexicons and reinforcement learning can occur by rewarding well-performing lexicons.

Outside the domain of peer review, we are interested in studying whether certain part of speech patterns indicate key information or exhibit strong sentiment within a document. This information can be extended into the area of text summarization. There are a number of potential applications where considerable time can be saved by filtering noise. These applications take two forms: single document extractive summarization (e.g., summarizing legal documents, terms and conditions statements, or attorney demands to include only the most relevant information) or multi-document abstractive summarization (e.g., surveying the most prominent political views of a politician’s constituents expressed on Twitter).

Online education is growing in both acceptance and practice. Any educators not convinced prior to the 2020 COVID-19 pandemic now see the usefulness, and occasional necessity, of instruction online. In such situations, and with an increase in the number of students, the need for scalable grading of complex assignments (e.g., essays, software projects, and designs) becomes obvious. Our work, which applies sentiment analysis to the meaningful content generated from a learning, peer-reviewing crowd facilitates efficient grading of open-ended assignments in large, online courses and promotes student engagement. Our domain-specific lexicon and sentiment scoring algorithm, coupled with our educational framework, equip an instructor to build a review form in a data-driven manner through aspect extraction and to generate a quantitative peer review comment grade from text alone. This process produces a host of supporting documents with information to increase an instructor’s 1) understanding of student opinion and 2) confidence in assigning a grade from peer review. It provides direct, specific, and timely feedback to both the instructor and students. Information is capital — we desire to gather and invest it in the classroom to make intelligent decisions that will ultimately benefit both students and instructors in their pursuit of excellence.
References


[95] Sasha Blair-Goldensohn, Kerry Hannan, Ryan McDonald, Tyler Neylon, George A Reis, and Jeff Reynar. Building a sentiment summarizer for local service reviews. In WWW workshop on NLP in the information explosion era, volume 14, pages 339–348, 2008.


Appendix A: Copyright Permissions

This appendix gives attribution and license information for figures used in this dissertation. Table A.1 lists the attribution information for each figure. The author created any figures not listed in this appendix.

- The Creative Commons Attribution 2.0 (CC BY 2.0) license details can be found at: https://creativecommons.org/licenses/by/2.0/legalcode

- The Creative Commons Attribution-ShareAlike 2.0 (CC BY-SA 2.0) license details can be found at: https://creativecommons.org/licenses/by-sa/2.0/legalcode

- The Creative Commons Attribution 3.0 (CC BY 3.0) license details can be found at: https://creativecommons.org/licenses/by/3.0/legalcode

- The Creative Commons Attribution-ShareAlike 3.0 (CC BY-SA 3.0) license details can be found at: https://creativecommons.org/licenses/by-sa/3.0/legalcode

- The Creative Commons Attribution-ShareAlike 4.0 (CC BY-SA 4.0) license details can be found at: https://creativecommons.org/licenses/by-sa/4.0/legalcode

- The Creative Commons Attribution-NoDerivatives 4.0 (CC BY-ND 4.0) license details can be found at: https://creativecommons.org/licenses/by-nd/4.0/legalcode
### Table A.1: Figure License Information

<table>
<thead>
<tr>
<th>Fig</th>
<th>Ch</th>
<th>Pg</th>
<th>Figure</th>
<th>Attribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>2</td>
<td>13</td>
<td><img src="image1.png" alt="Figure A" /> <img src="image2.png" alt="Figure B" /></td>
<td>“Main-qimg-48d5bd214e53d440fa32fc9e5300c894.png” by Sebastian Raschka is licensed under the CC BY-SA 4.0 license.</td>
</tr>
<tr>
<td>2.2</td>
<td>2</td>
<td>14</td>
<td><img src="image3.png" alt="Figure C" /></td>
<td>“Linear regression.svg” by Sewaqu is public domain.</td>
</tr>
<tr>
<td>2.3</td>
<td>2</td>
<td>15</td>
<td><img src="image4.png" alt="Figure D" /></td>
<td>“Colored neural network.svg” by Glosser.ca is licensed under the CC BY-SA 3.0 license.</td>
</tr>
<tr>
<td>Figure</td>
<td>2</td>
<td>16</td>
<td>&quot;Recursive Neural Tensor Network&quot; used with permission from Richard Socher.</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>---</td>
<td>----</td>
<td>--------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>Figure</td>
<td>3.1</td>
<td>28</td>
<td>&quot;Organic Learning&quot; used with permission from Marvel Lee Cummings.</td>
<td></td>
</tr>
<tr>
<td>Figure</td>
<td>3.3</td>
<td>30</td>
<td>&quot;Learning Process&quot; used with permission from Marvel Lee Cummings.</td>
<td></td>
</tr>
</tbody>
</table>
The permission below is for the reproduction of material in Chapter 2, Chapter 3, and Chapter 4, from *Designing Intelligent Review Forms for Peer Assessment: A Data-driven Approach* in Proceedings of the 2019 ASEE Annual Conference, Tampa, Fla. © 2019 American Society for Engineering Education.

In return for these rights, ASEE hereby grants the above author(s), and the employers for whom the work was performed, permission to:

- Reuse portions of the above paper in other works.
- Reproduce the above paper for personal or internal use, provided that (a) the source and ASEE copyright are indicated, (b) the copies are not used in a way that implies ASEE endorsement of product or service of an employer, and (c) the copies are not offered for sale.

The permission below is for the reproduction of material in Chapter 3, from *Ten Challenges in CAD Cyber Education* in Computer-Aided Design & Applications, 2018 VOL. 15, NO. 3, 432–442 © 2017 CAD Solutions, LLC and Chapter 5, from *A Domain-Dependent Lexicon to Augment CAD Peer Review* in Computer-Aided Design & Applications, 2020 VOL. 18, NO. 1, 186–198 © 2021 CAD Solutions, LLC.

**AUTHOR RIGHTS**

3. You hereby assert your moral rights to be identified as the author of the Article according to the UK Copyright Designs & Patents Act 1988.

4. You are permitted to use the material in the ways described in the Schedule of Author’s Rights providing that you meet all the conditions set out in the Schedule. These are rights which are personal to you and cannot be transferred by you to anyone else.

The permission below is for the reproduction of material in Chapter 7, from *Leveraging Peer Feedback to Improve Visualization Education* in PacificVis 2020, Tianjin, China. © 2020 IEEE.