DHQ: Digital Humanities Quarterly
2022
Volume 16 Number 2

Sentiment Analysis: Limits and Progress of the Syuzhet Package and Its Lexicons

Hoyeol Kim <elibooklover_at_gmail_dot_com>, Texas A&M University https://orcid.org/0000-0002-2049-7531

Abstract

Syuzhet is a dictionary-based tool for the sentiment analysis of literary texts that draws upon the Syuzhet, Bing, Afinn, and NRC lexicons. Syuzhet is a work in progress with the potential to become an invaluable tool for the sentiment analysis of literary texts. However, there have been doubts about sentiment analysis in the digital humanities field, especially after Swafford's impactful critique of Syuzhet. Since it is impossible to achieve 100% accuracy in sentiment analysis, we should embrace the imperfection and continue to use Syuzhet while also making efforts to fully understand its limits and abilities. In addition, we should continuously provide feedback for the tool, since the duty of improving digital tools belongs to all digital humanists who employ digital tools. This article explores the limits of and improvements made upon Syuzhet by examining and testing its code and functions with 19th century British novels; the subjectivity of its lexicons; and the validity of Swafford's critique.

1. Introduction

Text mining is no longer an uncommon research method when it comes to analyzing texts in the digital humanities. Once limited to the research field, text mining now influences "our lives, our teaching, and our scholarship, and digital humanists" [Binder 2016, 213] as "a logocentric practice." [Clement 2016, 534] Sentiment analysis, also known as opinion mining, shares common features with text mining when parsing, detecting, and locating words or sentences. Sentiment analysis is "the process of extracting an author's emotional intent from text." [Kwartler 2017, 85] Sentiment analysis has historically focused on product reviews, such as those of movies, hotels, cars, books, and restaurants, in addition to blog data, but current sentiment analysis has expanded to "stock markets, news articles, [and] political debates," [Medhat et al. 2014, 1094] and serves a variety of purposes. There have been attempts at employing sentiment analysis in literature, mainly grounded on lexicon-based approaches, but sentiment analysis in literature has been a target of attack in digital humanities due to its limits as a research method: Swafford's critique of the Syuzhet package made a great impact on the digital humanities field by alerting readers to the danger of choosing faulty tools, although her criticism rehashed already existing issues in sentiment analysis. Along with Swafford's critique of Syuzhet, other digital humanists shared erroneous results found through Syuzhet and expressed uneasy feelings about sentiment analysis in literature. [1] In reality, perfect codes/tools cannot exist, so we need to "embrace 'problems'" with Syuzhet "as a feature rather than a flaw" [Rhody 2015]. Ted Underwood asserts that if we "use algorithms in our research," we should "find out how they work." [Underwood 2014, 69] Similarly, when using digital tools, it is important to understand their functions, algorithms, and programming syntax, instead of simply drawing upon the visualized results, in order to avoid creating faulty results. 1

Sentiment analysis is a subfield of natural language processing, which classifies the sentiments of texts. Sentiment analysis researchers traditionally used lexicon-based and machine learning approaches. The machine learning approach uses machine learning algorithms with training datasets to classify sentiments based on linguistic features, whereas the lexicon-based approach draws upon the collection of precompiled sentiment lexicons to label words with sentiment scores. However, these traditional approaches revealed the limits of dealing with complex syntaxes and semantics. Recently, sentiment analysis researchers have proposed deep learning approaches such as transformers, cognition attention-based models, and sentiment-specific word embedding models. Deep learning approaches for sentiment analysis have been considered "as efficient methods due to their capability of learning the text without manual feature engineering" [Habimana et al. 2020] Traditional sentiment analysis approaches mainly drawing upon lexicons have around 70% accuracy, while recent deep learning approaches for sentiment analysis create state-of-the-art results. Sentiment analysis for literary texts, however, is still based on traditional approaches: Kim and Klinger note that "[i]t is true that much digital humanities research (especially dealing with text) uses the methods of text analysis that were in fashion in computational linguistic twenty years ago." [Kim and Klinger 2018, 18] Although sentiment analysis has been commonly employed in a variety of fields, mainly for commercial purposes, in addition to testing sentiment analysis with literary texts, 2 sentiment analysis for literature in the digital humanities is relatively new and received little attention until the Syuzhet package was first released, aimed at providing a proper tool for literary analysis. Syuzhet 0.2.0 was released on February 22, 2015 and was soon critiqued by Swafford, who pointed out problems with Syuzhet on her personal blog on March 2, 2015, such as (1) splitting sentences, (2) negators, (3) parts of speech, such as “well” and “like,” (4) lexicons being based on contemporary English words, (5) counting a word once for a sentence even if it is repeated, (6) scoring subjectivity, (7) satire and sarcasm, (8) foundation shapes [Swafford 2015]. Despite the effort by Jockers’ lab to create a useful tool for sentiment analysis tailored to analyzing literary texts, the limits of Syuzhet that Swafford pointed out caused digital humanists to have qualms about sentiment analysis in literature. After Swafford’s criticism against Syuzhet 0.2.0, Syuzhet 1.0.0 was released on April 28, 2016, followed by another release on December 14, 2017 of the 1.0.4 version. After almost three years since 1.0.4, Syuzhet 1.0.6 was released with minor updates on November 24. 2020.[2]

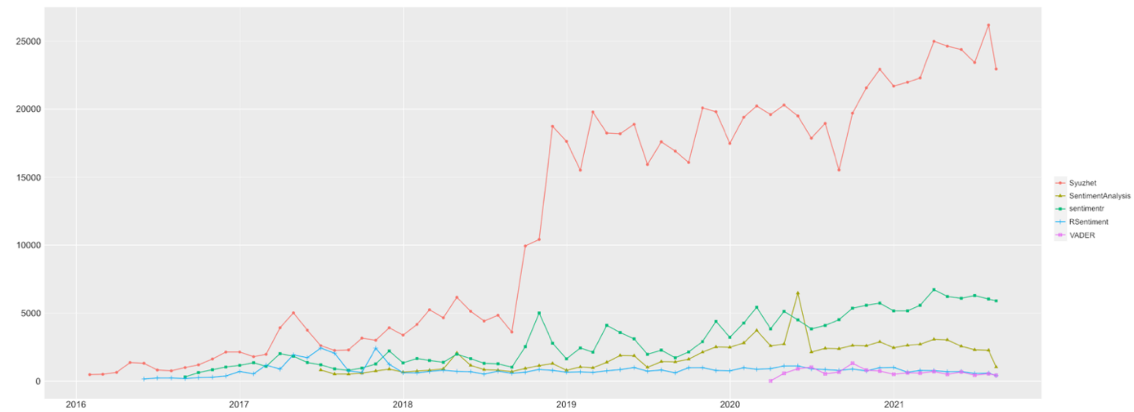

Figure 1. The number of monthly downloads for sentiment analysis R packages (created on August 17, 2021).

Figure 1 reveals that Syuzhet has been continuously downloaded as the most popular package for sentiment analysis in R.[3] In 2021, it has been downloaded more than 20,000 times monthly, but due to its limits, Syuzhet still remains difficult to validate as a research tool for sentiment analysis in the humanities. In the past, sentiment analysis researchers tested sentiment analysis with literary texts: Saif Mohammad [Mohammad 2012] created and tested the NRC lexicon with literary texts such as Shakespeare’s *Hamlet* and *As You Like It*, based on the basic emotion models of Ekman and Plutchik. Reagan et al. suggested the “six core emotional arcs” (rise, fall, fall-rise, rise-fall, rise-fall-rise, and fall-rise-fall) for fictional stories [Reagan et al. 2016]. Haider et al. [Haider et al. 2020] performed sentiment analysis with poems in English and German, using word embeddings as features and manually multi-labeling sentiments. Evgeny Kim and Roman Klinger [Kim and Klinger 2018] provided a survey of sentiment analysis in computational literary studies and examined the difficulties of detecting sentiments due to indirectly expressed emotions in literary texts. Michelangelo Misuraca et al. validated Syuzhet, using confusion matrices and macro-averaging with the course_evaluation dataset, of which each sentiment was manually labeled by Charles Welch and Rada Mihalcea [Welch and Mihalcea 2016]. In their test, the overall accuracy of Syuzhet was 0.671, and with the education dataset, the averages for precision, recall, and F-measure were 0.605, 0.526, and 0.526, respectively [Misuraca et al. 2020, 22]. Jockers asserts that “current benchmark studies suggest that [sentiment detection] accuracy” is “in the 70-80% range and that depends on genre” [Jockers 2015], but the accuracy of sentiment detection in the validation test of Syuzhet by Misuraca et al. was 67.1% [Misuraca et al. 2020, 22], which is a little lower than the 70-80% range Jockers argued to defend Syuzhet. 3

Despite the low accuracy of Syuzhet, it is one of the most popular sentiment analysis tools for R, as Figure 1 shows. After the criticism against Syuzhet, it was difficult to find new sentiment analysis research in the digital humanities, although Syuzhet users have drastically increased in the meantime. The problem is that sentiment analysis tools in R heavily draw upon lexicons, which are far from deep learning approaches in regards to methodology. Recently, despite the criticism against Syuzhet, which resulted in digital humanists having reservations towards sentiment analysis as a research method in the humanities, there were a couple of digital humanists who presented at the ACH2021 conference about sentiment analysis in the humanities using VADER (Valence Aware Dictionary and sEntiment Reasoner) for sentiment analysis with humanities data. VADER is a lexicon and rule-based sentiment analysis tool tailored to the sentiment analysis of social media. As Stéfan Sinclair, Stan Ruecker, and Milena Radzikowska emphasize, cultivating a sufficient understanding of digital tools is important since “the interpretive work is being guided and biased by the data and software” [Sinclair et al. 2013, ¶54]. While Syuzhet has been controversial as a research method due to its limits, it is still meaningful for helping literary critics grasp what they should consider when performing sentiment analysis. 4

Therefore, I decided to closely examine Syuzhet 1.0.6 to impart the limits and progress of Syuzhet, with the subjects of my experiment being mainly from 19th century British novels, since they are not under copyright, are long enough to produce valid analyses, and are credited for their well-structured plots. I begin by exploring similar and dissimilar results of sentiment plots, the similarity of deciding positivity and negativity between the lexicons, and the percentage of shared words between lexicons with four lexicons for sentiment analysis: Syuzhet, Bing, Afinn, and NRC. As there are currently no validation datasets for the sentiment analysis of Victorian fiction, I examine the results of sentiment analysis with Charles Dickens's *Our Mutual Friend*, George Eliot's *Middlemarch*, and Charlotte Brontë's *Jane Eyre*. I conclude that Syuzhet needs to be improved in order to capture semantic and syntactic information, that the usage of DCT (Discrete Cosine Transformation) for sentiment analysis plots creates distorted results. Finally, I suggest that we should use deep learning approaches for sentiment analysis in the humanities. 5

2. Lexicons

The term Syuzhet stems from "the Russian Formalists Victor Shklovsky and Vladimir Propp who divided narrative into two components, the 'fabula' and the 'syuzhet'" to depict narrative structures of story. Syuzhet intends to provide "the latent structure of narrative by means of sentiment analysis" and specifically "the emotional shifts that serve as proxies for the narrative movement between conflict and conflict resolution." [Jockers 2017b] Jockers' explanation of Syuzhet describes it as a sentiment analysis tool for the analysis of literary texts. Syuzhet is a lexicon-based package, mainly drawing upon four standard lexicons: Syuzhet, Bing, Afinn, and NRC. 6

	Syuzhet	Bing	Afinn	NRC
No. of Positive Words	3587	2006	878	2312
No. of Negative Words	7161	4783	1598	3324
No. of Other Words	-	-	1	8265
Total	10748	6789	2477	13901

Table 1. Number of Sentiment Words in Lexicons Used in the Syuzhet Package

The Bing, Afinn, and Syuzhet lexicons provide polarity which sorts words into positive or negative positions with numeric values. The Bing lexicon[4] has a binary categorization, which simply has two values of –1 and 1. The Afinn lexicon [5] grades words between –5 and 5. The Syuzhet lexicon has more specific values for each sentiment word, ranging between –1 and 1, which are –1.0, –0.8 –0.75, –0.6, –0.5, –0.4, –0.25, 0.1, 0.25, 0.4, 0.5, 0.6, 0.75, 0.8, 1.0. The NRC lexicon[6] sorts sentiment words into categories consisting of positive, negative, anger, anticipation, disgust, fear, joy, sadness, surprise and trust. The other words from the NRC lexicon in Table 1 consist of anger (1247), anticipation (839), disgust (1058), fear (1476), joy (689), sadness (1191), surprise (534), and trust (1231). A number of words from the NRC lexicon are included in different categories at the same time, but the Syuzhet package can only work with positive and negative lexicons from the NRC lexicon. Excluding duplicate words in the different feeling categories of the NRC lexicon, there are 6,468 unique words. Among these, there are 81 words which belong to both positive and negative categories, such as "boisterous," "endless," and "revolution." The Syuzhet package processes those 81 words with a score of 0. In addition, if a word was not categorized as positive or negative, it will score 0. For example, "confront" falls into two categories: anger and anticipation, but scores 0, whereas "annoy" scores –1, which is categorized as negative, anger, and disgust in the NRC lexicons. 7

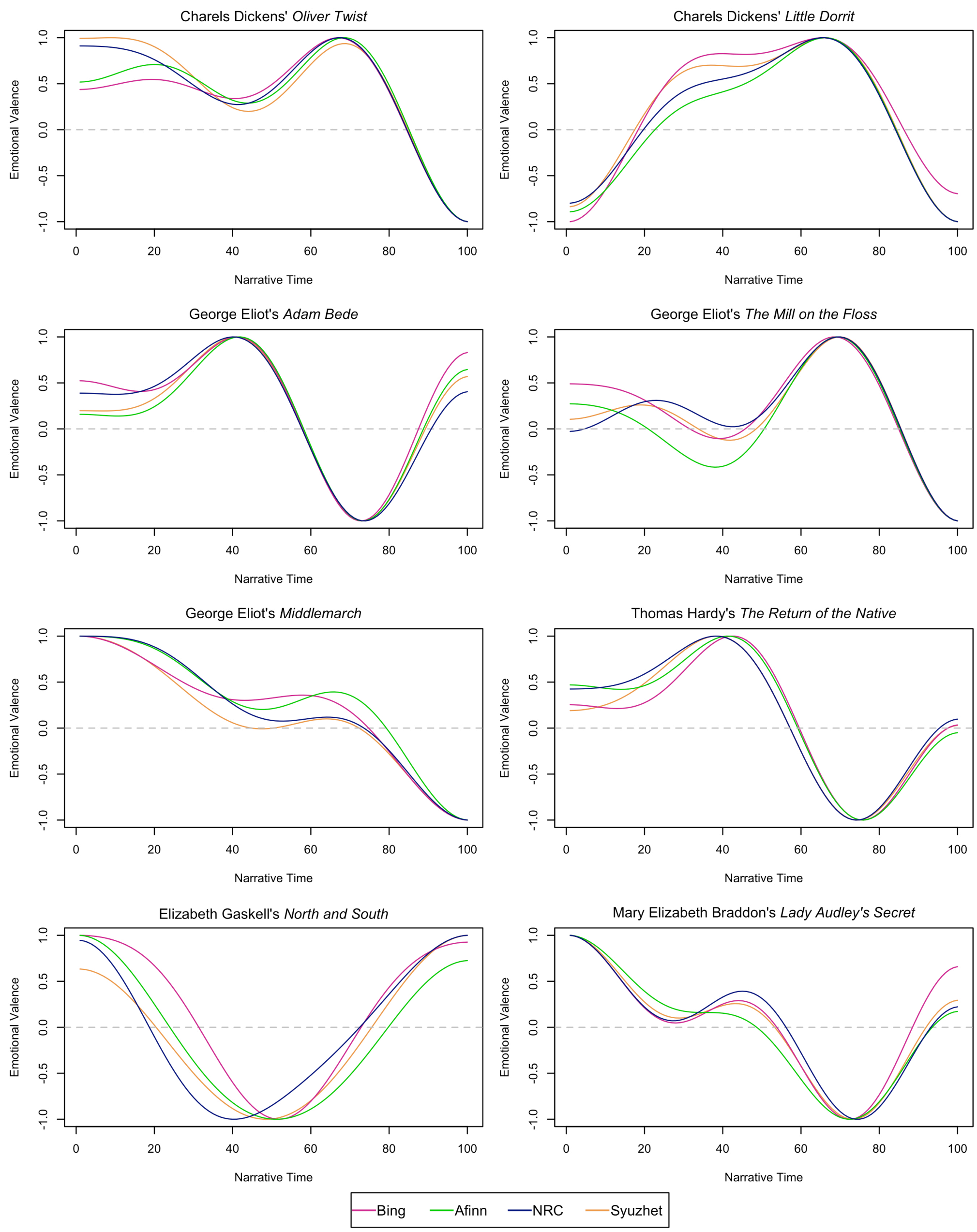

Figure 2. Similar results from four different lexicons

Figures 2 and 3 were created through the get_dct_transform function of Syuzhet using four different lexicons, Bing, Afinn, NRC, and Syuzhet, for sixteen novels. In figure 2, the emotional valence of each lexicon is similar over the narrative time from eight novels: Charles Dickens's *Oliver Twist* and *Little Dorrit*, George Eliot's *Adam Bede*, *The Mill on the Floss* and *Middlemarch*, Thomas Hardy's *The Return of the Native*, Elizabeth Gaskell's *North and South*, and Mary Elizabeth Braddon's *Lady Audley's Secret*.

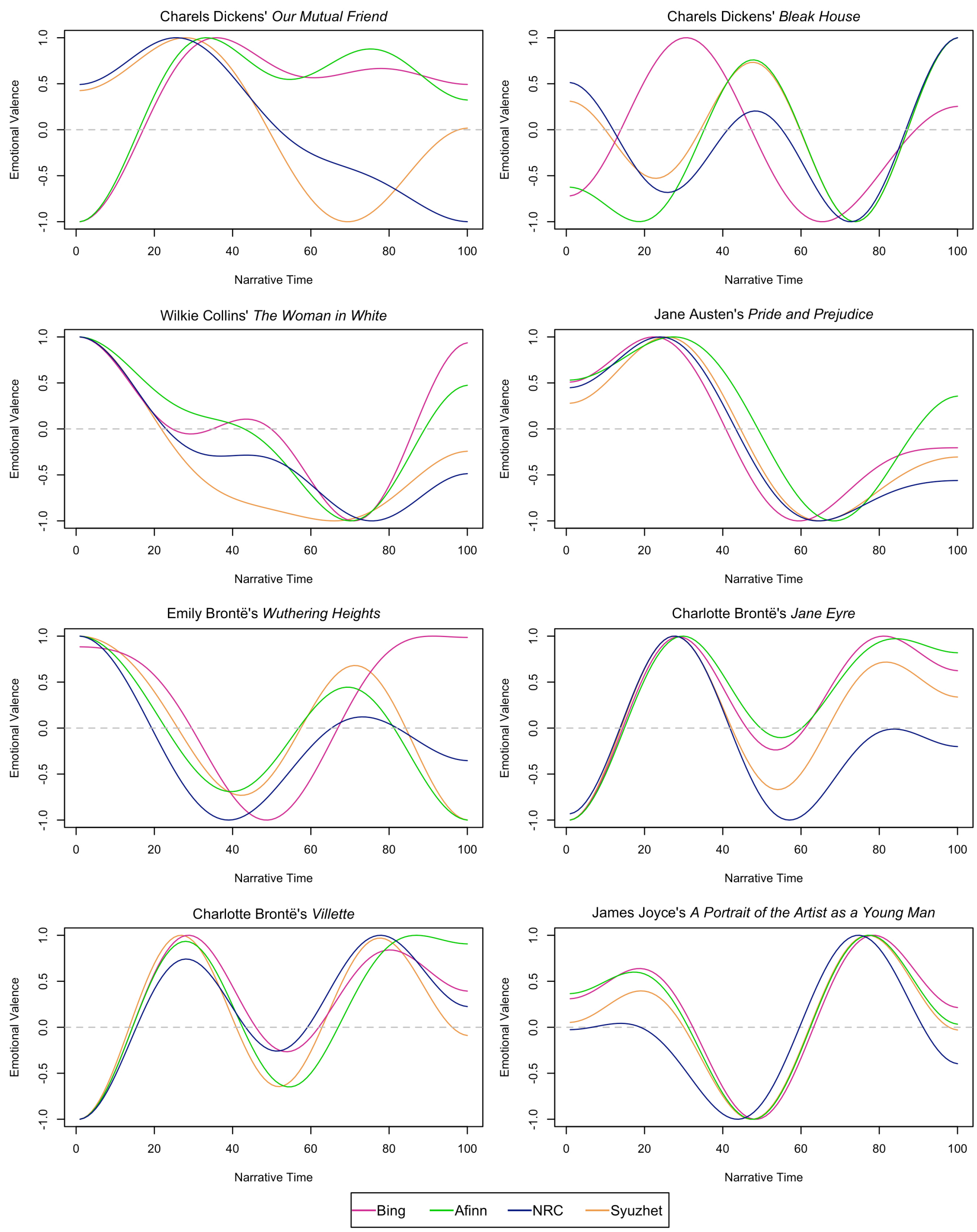

Figure 3. Figure 3: Differing results from four different lexicons

Figure 3, however, reveals inconsistent emotional valences from four lexicons for eight novels: Charles Dickens's *Our Mutual Friend* 9
and *Bleak House*, Wilkie Collins' *The Woman in White*, Jane Austen's *Pride and Prejudice*, Emily Brontë's *Wuthering Heights*, Charlotte Brontë's *Jane Eyre* and *Villette*, and James Joyce's *A Portrait of the Artist as a Young Man*.

What causes different sentiment analysis results to be generated depending on the lexicon? I examined the differences between 10
the four lexicons based on positivity and negativity in order to find the reasons why sentiment trajectories could be different between them. Table 2 reveals that the Bing and Afinn lexicons have the highest similarity of deciding positivity and negativity, whereas the Syuzhet and NRC lexicons have the lowest number between the results, although the number is still high.

Lexicons (No. of Words)	Syuzhet-Bing (5,910)	Syuzhet-Afinn (2,285)	Syuzhet-NRC (4,783)	Bing-Afinn (1,315)	Bing-NRC (2,396)	Afinn-NRC (990)
Similarity of Deciding Positivity and Negativity	98.26%	98.47%	96.59%	98.71%	98.33%	98.18%

Table 2. Similarity of deciding positivity and negativity between lexicons used in the Syuzhet package

The percent similarity for giving the same words positive or negative values between two different lexicons are the followings: Syuzhet-Bing (98.26%, 5,910 words), Syuzhet-Afinn (98.47%, 2,285 words), Syuzhet-NRC (96.59%, 4,783 words), Bing-Afinn (98.71%, 1,315 words), Bing-NRC (98.33%, 2,396 words), and Afinn-NRC (98.18%, 990 words). This means that the Syuzhet and Bing lexicons have 5,910 common words that, when given positive and negative scores, conflict 1.74% of the time. For example, the words “avenge,” “enough,” and “envy” are scored 0.25, –0.25, and –0.8 by the Syuzhet lexicon, versus –1, 1, and 1 by the Bing lexicon. Looking into the comparison of the Syuzhet and NRC lexicons, the words “absolute,” “ancient,” and “blush” score –0.25, 0.25, and 0.6 in the Syuzhet lexicon, versus 1, –1, and –1 in the NRC lexicon, respectively. These different decisions whether words will be assigned positive or negative can bring about different results during sentiment analysis, as shown in figure 3. 11

Lexicons (No. of Words)	Syuzhet-Bing (5,910)	Syuzhet-Afinn (2,285)	Syuzhet-NRC (4,783)	Bing-Afinn (1,315)	Bing-NRC (2,396)	Afinn-NRC (990)
Syuzhet	54.99%	21.26%	44.50%	-	-	-
Bing	87.05%	-	-	19.37%	35.29%	-
Afinn	-	92.25%	-	53.09%	-	39.97%
NRC	-	-	73.95%	-	37.04%	15.31%

Table 3. Percentage of shared words between lexicons used in the Syuzhet package

Based on table 3, the percentage of words included in the Syuzhet package that are shared with any given lexicon are relatively low across the board. This is most likely due to the fact that the Syuzhet lexicon was created much later with reference to the Bing, Afinn, and NRC lexicons, and therefore includes words from all three. Because of this, Syuzhet has the most words of any lexicon (not including repeated words in the NRC lexicon) at 10,748 words, causing the disproportion between the percentages of shared words for Syuzhet and the lexicons it is being compared with. Similarly, Afinn, with the fewest words of the four lexicons, when compared with them generates higher percentages for itself. 12

Despite having the same tool setting conditions, depending on the lexicon, sentiment trajectories could be different due to the subjectivity of the lexicons. The inconsistent sentiment scores of the Syuzhet lexicon result in the discrediting of lexicon-based sentiment analysis. Stephen Ramsay states that literary criticism is not only “a qualitative matter” but also “an insistently subject manner of engagement.” [Ramsay 2011] Likewise, creating lexicons is a “subject manner of engagement” [Ramsay 2011, 8] through the subjective interpretation of emotions used in labeling words with scores. Sentiment analysis packages provide customizing functions, either through the customization of dictionaries or the use of dictionaries that are created from scratch, in order to overcome this limit. Nonetheless, it would be challenging to create a dictionary that avoids every critique of subjectivity. 13

Syuzhet 1.0.6 has not provided a function to use custom dictionaries yet. Syuzhet 2.0.0 is expected to provide the function, but it usually requires a considerable amount of time and effort to create sentiment dictionaries, and customized dictionaries might face the question of reliability and credibility when used in research. Instead of creating a sentiment dictionary from scratch, researchers can use pre-made sentiment dictionaries, such as the psychological Harvard-IV dictionary [7] (DictionaryGI), or customize their sentiment analysis by adding algorithms, but they cannot change the sentiment scores from existing lexicons. 14

3. Syuzhet

3.1 Parsing

The goal of opinion mining is to generate relevant information from texts for analysis. To do so, parsing text is the first step. However, there can be distortions in the process of text mining if raw data are not trimmed. Therefore, well-structured text data need to be inputted for sentiment analysis to generate the correct data. In Syuzhet, there are two different ways to parse text and transform it into vector values: (1) Tokenizing the text into sentences, and then transforming the text into a numeric vector for each sentence. (2) Tokenizing the text into words, and then transforming each word into vector representations. Depending on the 15 purpose of research, the text is tokenized into sentences or words through either the get_sentences function or the get_tokens function. For the sentiment analysis of novels, the first method, which tokenizes the text into sentences, is normally chosen, so I will focus on parsing the text into sentences using the Syuzhet package. The Syuzhet package originally (versions 1.0.1 and earlier) called upon the OpenNLP [8] API, which is an open source, in order to implement the get_sentences and the get_tokens functions. In addition, the Syuzhet package originally required installing Oracle's Java and two R packages, namely "openNLPdata" and "rJava," in order to use the OpenNLP parser, which was not user-friendly. Both the get_sentences function and the get_tokens function parse sentences or tokenize words into numeric vectors of sentiment values. Parsing text is a basic query used to process natural languages, as computers cannot read characters, only numbers. Swafford points out the problems with the OpenNLP parser when grouping sentences [Swafford 2015], and Jockers responds to her by asserting that the OpenNLP parser and the Stanford CoreNLP parser are "good enough" [Jockers 2015], although he admits that these parsers have problems. In fact, the Stanford parser [9] is a well-constructed tool, which applies a Part-of-Speech (POS) tagging. The OpenNLP parser has been improved, but I found that Syuzhet no longer uses the OpenNLP parser for the get_sentences function, despite Jockers mentioning that it does [Jockers 2017a]. Instead, Syuzhet draws upon the Textshape package developed by Tyler Rinker for parsing sentences. It seems the Syuzhet manual has not been updated yet, as this change in the parser by Jockers went undocumented. It is possible that Jockers made the change in order to acknowledge the limits of the OpenNLP parser for literary text. Syuzhet 1.0.2 was updated with the removal of the Java dependency, which means that Syuzhet users do not have to install Oracle's Java and its dependent packages, "openNLPdata" and "rJava," anymore to utilize the Textshape package, in addition to parallelization of the get_sentiment function by Philip Bulsink on July 28, 2017.

Author	**Title**	**Syuzhet 0.2.0**	**Syuzhet 1.0.6**	**Change**
Charles Dickens	*Oliver Twist*	6,887	9,128	+32.54%
	Bleak House	18,171	20,319	+11.82%
	Little Dorrit	16,241	18,110	+11.51%
	Our Mutual Friend	15,339	20,261	+32.09%
George Eliot	*Adam Bede*	8,199	8,909	+8.66%
	Mill on the Floss	7,957	8,768	+10.19%
	Middlemarch	13,540	14,415	+6.46%
Charlotte Brontë	*Jane Eyre*	8,605	9,663	+12.30%
	Villette	9,172	10,179	+10.98%
Emily Brontë	*Wuthering Heights*	5,528	6,755	+22.20%
Jane Austen	*Pride and Prejudice*	5,633	5,938	+5.41%
Wilkie Collins	*The Woman in White*	12,675	13,472	+6.29%
Elizabeth Gaskell	*North and South*	8,739	10,418	+19.21%
Mary Elizabeth Braddon	*Lady Audley's Secret*	6,670	7,288	+9.27%
Thomas Hardy	*The Return of the Native*	7,888	8,922	+13.11%
James Joyce	*A Portrait of the Artist as a Young Man*	5,146	5,347	+3.91%
Total Sentences		**156,390**	**177,892**	**+13.75%**

Table 4. Comparison of the parsing results from sixteen novels using Syuzhet 0.2.0 and 1.0.6

In table 4, I compared the parsing results from sixteen novels using Syuzhet 0.2.0 with the OpenNLP parser and Syuzhet 1.0.6 with the Textshape parser in order to examine the improvements of the parsing function in Syuzhet. Table 4 reveals the fact that the parsing function of Syuzhet was improved across the board after Syuzhet deployed the Textshape package for parsing instead of the OpenNLP parser. The parsing results from the sixteen novels between Syuzhet 0.2.0 and 1.0.6 have a 13.75% increase. For example, table 5, which shows the parsing result from Charles Dickens's *Our Mutual Friend*, informs that the parsing function of Syuzhet 1.0.2 was improved by splitting sentences more correctly. The OpenNLP parser often failed to split sentences such as: "'I'll take the rest of the spell.' 'No, no, father!'" [Dickens 1952] In addition, the OpenNLP parser did not split sentences which ended with exclamation and quotation marks. For example, table 5, which is the parsing result from George Eliot's *Middlemarch*, is one of examples that proves that the OpenNLP does not process an exclamation mark as a splitter. In other words, the Textshape package parsed the text into sentences more correctly than the OpenNLP parser based on tables 4, 5 and 6. 16

Syuzhet ≤ 1.0.1	**Syuzhet ≥ 1.0.2**
"Has Mr. Casaubon a great soul?" Celia was not without a touch of naive malice.	"Has Mr. Casaubon a great soul?"
	Celia was not without a touch of naive malice.

Table 5. Parsing from George Eliot's *Middlemarch* (Chapter 1)

Syuzhet ≤ 1.0.1	**Syuzhet ≥ 1.0.2**
'Here! and give me hold of the sculls.	'Here!
	and give me hold of the sculls.
I'll take the rest of the spell.' 'No, no, father!	I'll take the rest of the spell.'
	'No, no, father!
No! I can't indeed.	No!
	I can't indeed.

Table 6. Parsing from Charles Dickens's *Our Mutual Friend* (Book 1, Chapter 1)

Lastly, I tested the parsing function of Syuzhet with Charles Dickens's *Bleak House* to compare the part of chapter 3 where 17
Swafford pointed out grouping errors (see table 6).

Syuzhet ≤ 1.0.1	**Syuzhet ≥ 1.0.2**
"Mrs. Rachael, I needn't inform you who were acquainted with the late Miss Barbary's affairs, that her means die with her and that this young lady, now her aunt is dead — " "My aunt, sir!" "It is really of no use carrying on a deception when no object is to be gained by it," said Mr. Kenge smoothly, "Aunt in fact, though not in law."	"Mrs. Rachael, I needn't inform you who were acquainted with the late Miss Barbary's affairs, that her means die with her and that this young lady, now her aunt is dead — " "My aunt, sir!"
	"It is really of no use carrying on a deception when no object is to be gained by it," said Mr. Kenge smoothly, "Aunt in fact, though not in law."

Table 7. Parsing from Charles Dickens's *Bleak House* (Book 1, Chapter 3)

Based on the parsing result in table 7, the Textshape parser split sentences after an exclamation mark, but not a dash. Syuzhet 18
1.0.6 with the Textshape parser sorts sentences better than Syuzhet 1.0.1 with the OpenNLP parser. The Textshape parser, however, still has room for improvement for splitting sentences. For example, the Textshape parser infrequently fails to split sentences based on a period, such as: "'My dear, I don't know it,' said I. 'You do,' she said very shortly." [Dickens 1948, Book 1, Chapter 4] in addition to the dash. Based on the parsed result of sixteen novels, I concluded that the Textshape package basically separates sentences based on a period, exclamation mark, or question mark.

3.2 Comparison of Sentiment Values

Syuzhet allocates different numeric vectors to each word/sentence based on the lexicon chosen. These transitioned numeric 19
vectors are turned into structured data or visualization for further analysis. In Syuzhet, there are four different functions to show the emotional valence of stories throughout narrative time: get_sentiment, get_percentage_values, get_transformed_values, and get_dct_transform. The get_percentage_values, get_transformed_values and get_dct_transform functions are percentage-based functions, whereas the get_sentiment function is based on the number of sentences. The get_sentiment function transforms texts into accumulative numeric values for sentiment analysis by matching each word with sentiment scores in selected lexicons. The get_percentage_values function "divides a text into an equal number of 'chunks' and then calculates the mean sentiment valence for each."[Jockers 2017a] The get_transformed_values function uses the Fourier with a low pass filter to make the graph smooth, but Jockers recommends get_dct_transform in lieu of get_transformed_values because get_transformed_values is only being maintained for legacy purposes. The get_dct_tansform function draws upon "the simpler discrete cosine transformation (DCT)," and its strength is to depict "edge values in the smoothed version of the sentiment vector" [Jockers 2017a]. DCT is mostly used in digital media to efficiently process calculations and compress digital media, but it can create errors between data blocks. The fundamental idea of DCT is to compress data for efficiency by removing noise, but in doing so, DCT can distort the original data when performing sentiment analysis.

I tested Syuzhet (1.0.6), SentimentAnalysis (1.3-4), sentimentr (2.7.1), RSentiment (2.2.2), and VADER (R, 0.2.1) with seven different sentences to see how each lexicon-based sentiment analysis tool generates sentiment scores (see table 8). SentimentAnalysis utilizes lexicons such as QDAP (Quantitative Discourse Analysis Package) dictionary, GI (Havard-IV) dictionary, and LM (Loughran-McDonald) dictionary. sentimentr by default uses the combination of an augmentation version of the Syuzhet and Bing lexicons. Similarly, RSentiment uses the Bing lexicon, whereas VADER deploys its own lexicon. 20

	Sentences	Syuzhet				SentimentAnalysis[10]			sentimentr	RSentiment	VADER[11]
		Syuzhet	Bing	Afinn	NRC	QDAP	GI	LM	Syuzhet & Bing	Bing	VADER
A.	She was happy.	0.75	1	3	1	1	1	1	0.433	1	0.572
B.	She was not happy.	0.75	1	3	1	1	1	1	–0.375	–1	–0.458
C.	She was sad.	–0.5	–1	–2	0	–1	–1	0	–0.288	–1	–0.477
D.	She was happy but she is sad now.	0.25	0	1	1	0	0	0.333	–0.397	0	–0.421
E.	She was happy, and she is still happy now.	0.75	1	3	1	0.5	0.5	0.5	0.562	2	0.813
F.	She was happy but she is no longer happy.	0.75	1	2	1	0.666	0.666	0.666	–0.562	0	–0.391
G.	She was extremely happy.	0.75	1	3	1	0	0.5	0.5	0.675	1	0.611

Table 8. Experiment in Syuzhet, SentimentAnalysis, sentimentr, RSentiment, and VADER with lexicons

The sentiment scores of each sentence created with Syuzhet are positive, aside from C. I tested B by replacing “not” with “never,” and I got the same result with Syuzhet. Furthermore, C produced –0.5 points, and D generated 0.25 points. The word, “sad” was given –0.5 points. D scored 0.25 points due to the combination of “sad” (–0.5) and “happy” (0.75). This result indicates that Syuzhet still has issues when semantically detecting sentences, as Swafford has pointed out in the Syuzhet 0.2.0 version. The comparison between A and B shows that Syuzhet has no function to detect negators. D and F depict the lack of a detector for adversative conjunctions in Syuzhet. In addition, the fact that the sentiment score of A is the same with that of G reveals that Syuzhet does not properly detect amplifiers. Table 8 demonstrates how Syuzhet simply reports accumulative sentiment scores based on the words in each sentence, as does SentimentAnalysis, while VADER and sentimentr employ detectors for negators, adversative conjunctions, and amplifiers. 21

VADER and sentimentr provide functions for detecting negators (not, aren’t, no), amplifiers (really, absolutely, very), de-amplifiers (hardly, barely, rarely), and adversative conjunctions/transitions (nonetheless, however, although). Due to the development of machine learning algorithms, dealing with negators is no longer the challenge it used to be. Negators in sentences can be detected and processed through n-grams with high-orders based on supervised algorithms [Jung et al. 2008]. Rinker, who developed sentimentr, asserts that negators appear in conjunction with about 20% of polarized words in a sentence. Rinker created valence shifters based on n-grams with high-orders to deal with negators, amplifiers, de-amplifiers, and adversative conjunctions/transitions. Through valence shifters, the accuracy of sentiment analysis has improved, though sentimentr still creates inconsistent results based on the total number of tokens. For example, the sentiment values between the three sentences, “She isn’t happy” (–0.433), “She is not happy” (–0.375), and “Today, she is not happy” (–0.335) are different. 22

Current sentiment analysis tools still need to improve through alternative approaches. Lexicon-based sentiment analysis has “the inability to find opinion words with domain and context specific orientations.” [Medhat et al. 2014, 1102] The layers of abstraction must be deeper to semantically and syntactically detect sentences in lexicon-based sentiment analysis tools, which simply 23 transform sentiment words into numeric vectors based on sentiment lexicons, then create visualizations to depict the data. Likewise, Syuzhet still fails to properly deal with negators, amplifiers, de-amplifiers, and adversative conjunctions/transitions.

4. Sentiment Analysis of Charles Dickens's *Our Mutual Friend*, George Eliot's *Middlemarch*, and Charlotte Brontë's *Jane Eyre* through Syuzhet

I selected the Syuzhet lexicon to test four different functions with Charles Dickens's *Our Mutual Friend*, George Eliot's *Middlemarch*, and Charlotte Brontë's *Jane Eyre* in order to examine the compatibility, as well as the limits, of Syuzhet with literature. In figure 4, each function depicts the emotional valence of *Our Mutual Friend* in different ways. Regarding the settings of the get_transformed_values and get_dct_transform functions, scale_vals=FALSE and scale_range=TRUE. The plot trajectory created by the get_sentiment function is complicated and condensed, showing both positive and negative emotion. Nonetheless, it is a useful function when it comes to meticulously grasping sentiment flow in a story. 24

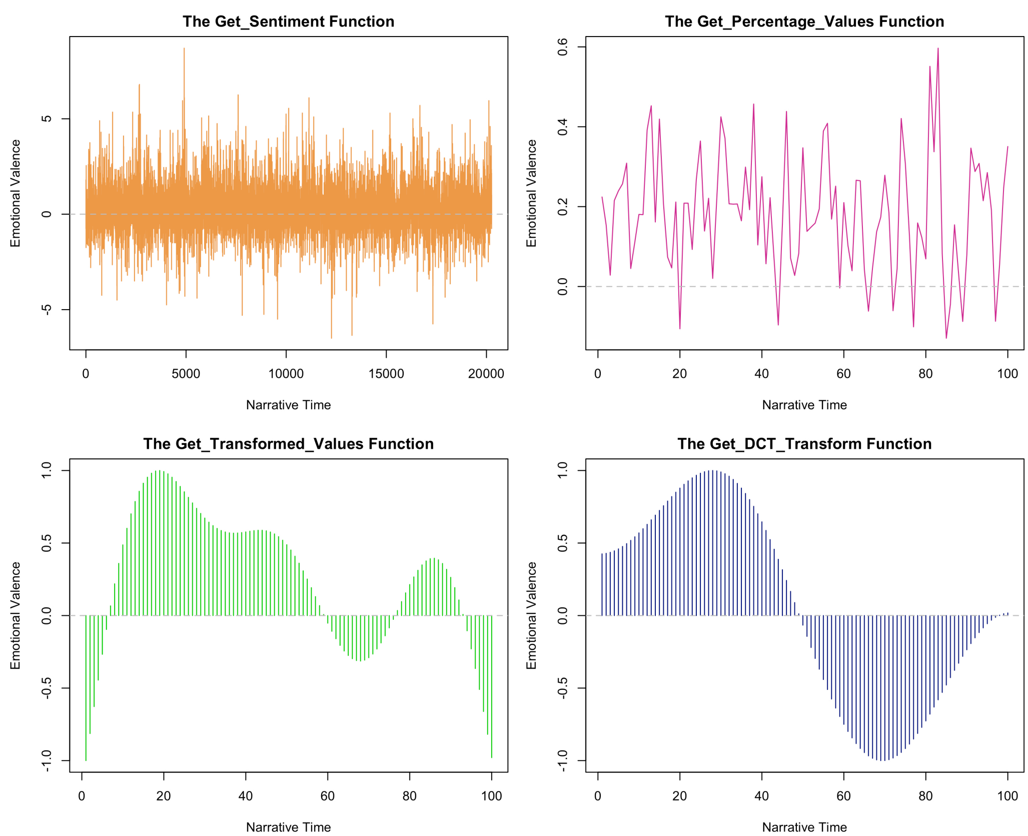

Figure 4. Comparison of four different functions based on the Syuzhet lexicon from Charles Dickens's Our Mutual Friend

Looking into the raw file after it was processed by the get_sentiment function using the Syuzhet lexicon, 7,167 sentences out of 20,261 sentences scored 0 (neutral), the number of positive sentences was 8,123, and the number of negative sentences was 4,971. The positive average was 0.95, and the negative average was –0.81. Based on the emotion trajectories created by the get_sentiment and get_percentage_values functions, the whole plot of *Our Mutual Friend* is swayed by positive feelings except for eight chapters. The get_sentiment result shows that each chapter entails both positive and negative emotions, and that overall, positive sentiment governs over negative feelings. The get_percentage_values function reveals that there are more negative feelings expressed in books 3 and 4. The highest score (8.7) is found in the last chapter of book 1, x=4907: "My Dear Sir,–Having consented to preside at the forthcoming Annual Dinner of the Family Party Fund, and feeling deeply impressed with the immense usefulness of that noble Institution and the great importance of its being supported by a List of Stewards that shall prove to the public the interest taken in it by popular and distinguished men, I have undertaken to ask you to become a Steward on that occasion." [Dickens 1952] The results from the get_dct_transform function reveal that *Our Mutual Friend* begins with slightly positive feelings, then reaches a peak of positivity in book 2, before reversing into negativity from book 3. This makes sense, as in book 2, there are a number of jocund and cheerful events, such as Mr. Headstone's and Mr. Eugene Wrayburn's wooing towards 25

Lizzie, Mr. Veneering's luxurious life, Mr. and Mrs. Lammle's social life, Fledgeby's smooth business, Mr. Boffin's purchase of an old mansion, and Bella's taste for money. The lowest score (–6.5), on the other hand, is found in book 3 chapter 8, x=12262: "This boastful handiwork of ours, which fails in its terrors for the professional pauper, the sturdy breaker of windows and the rampant tearer of clothes, strikes with a cruel and a wicked stab at the stricken sufferer, and is a horror to the deserving and unfortunate." [Dickens 1952] The get_dct_transform function reveals the dominance of negative feelings in the novel from the halfway point, though it becomes positive once more in the ending. Similarly, between x≈10000 and x≈15000 (Book 3) from the get_sentiment function, high values of negative sentiment are often found. Emotions fluctuate in book 3, but the negative atmosphere is dominant in book 3 due to an endless string of troubling plots such as Lizzie's disappearance and return, Mr. Riderhood's drowning, Bella's conflicts about money, Silas Wegg's plot, Headstone's jealousy, Mr. and Mrs. Lammle's bankruptcy, and Mr. Boffin's anger over Rokesmith. Although chapter 4 is filled with a positive ambience surrounding Mr. and Mrs. Wilfer's wedding anniversary, the emotional flows of the plot shown by the get_dct_transform function are relatively correct. Still, it is impossible to assert that the get_dct_transform function is 100% correct due to its over-simplification of emotion flows, the inconsistent values of lexicons, and the absence of functions which detect negators, amplifiers, de-amplifiers, and adversative conjunctions/transitions. For example, at x≈20, sentiment is extremely negative in the get_percentage_values function, whereas both the get_transformed_values and get_dct_transform functions have positive values, which are erroneous results caused by the smoothing filter occurring in their functions.

For the first 8% of narrative time, sentiment values are opposite between the get_percentage_values and get_transformed_values functions, with positive and negative scores respectively (figure 4). Here, the get_transformed_values function does not correctly reveal the sentiment trajectories compared to the other functions. As I mentioned above, Jockers does not recommend use of the get_transformed_values function, which has been preserved for legacy purposes, but it should be referenced since the get_dct_transform function derives from the get_transformed_values function. The distinctive difference between the two functions is low pass size. The get_transformed_values and the get_dct_transform functions have low pass sizes of 2 and 5 respectively, which denotes that the get_dct_transform function simplifies sentiment trajectory more than the get_transformed_values function does.

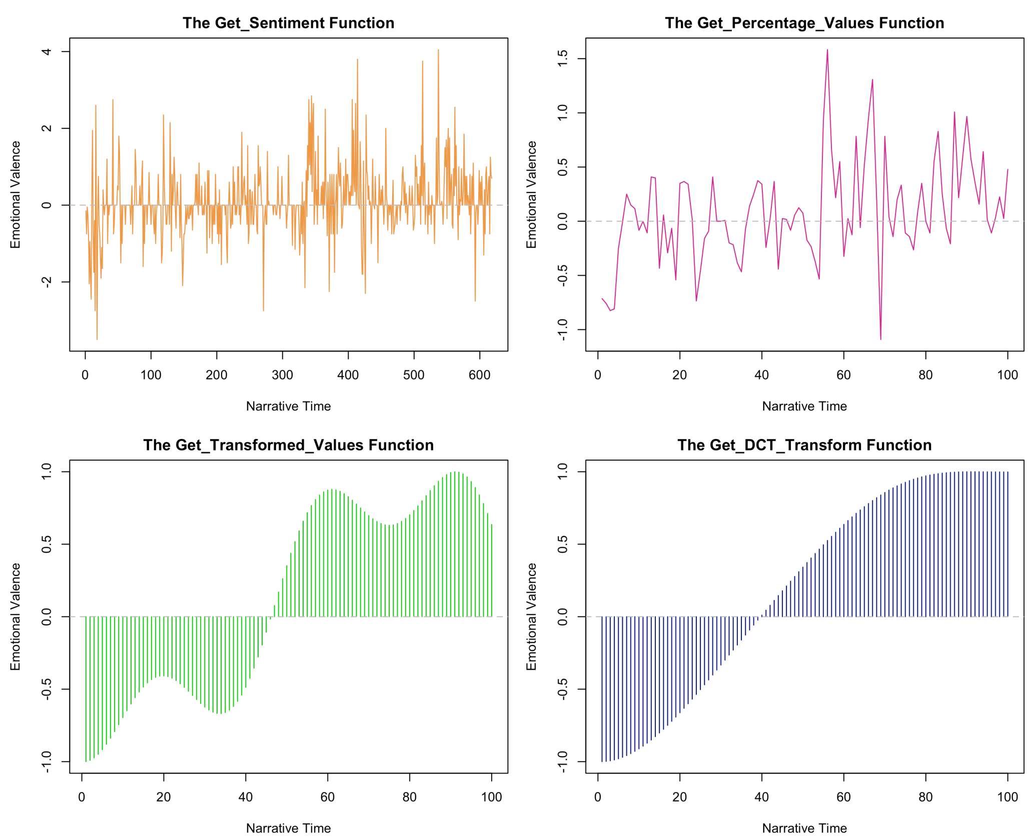

Figure 5. Comparison of four different functions from Book 4, Chapter 15 and 16 of *Our Mutual Friend*

In order to specifically examine the sentiment aspect from figure 4, I chose chapters 15 and 16, both from book 4, which are from x≈96% (19499) to x≈99% (20116) in figure 4. After parsing, chapters 15 and 16 consist of 336 and 282 chunks, respectively. Therefore, in figure 5, chapter 15 is between x=0% and x≈54%, and the rest is chapter 16. Looking into the raw file after it was processed by the get_sentiment function with the Syuzhet lexicon, 154 and 96 sentences in chapter 15 and 16, respectively, scored 0 (neutral), 68 and 129 sentences had positive values, and 114 and 57 sentences recorded negative values. Although the number of sentences in chapter 15 and 16 combined is less than 1000, which might bring about incorrect results, the four visualizations in figure 5 appear to appropriately demonstrate the two chapters. Chapter 15 is comprised of Riderhood's blackmail towards Headstone and their subsequent death in the river. The scene which depicts Riderhood staying in Headstone's classroom is filled with tension, and the result of Syuzhet reflects this with negative sentiment values, the lowest of which is (–3.5): "But, not to be still further defrauded and overreached–which he would be, if implicated by Riderhood, and punished by the law for his abject failure, as though it had been a success–he kept close in his school during the day, ventured out warily at night, and went no more to the railway station."[Dickens 1952] In addition, negative feelings are dominant due to Headstone's attempt to drown Riderhood, which results in both of their deaths, and which occurs in the last twenty sentences in chapter 15. 27

Nonetheless, the foundation shapes created by the get_transformed_values and get_dct_transform functions depict positive spikes, whereas the trajectories created by the get_sentiment and get_percentage_values functions at x=317 (x≈51%), to x=336 (x≈54%) correctly show negative spikes. The foundation shapes of Syuzhet, due to its smoothing feature, do not properly handle the drastic sentiment changes from the end of chapter 15, which describes drowning–"When the two were found, lying under the ooze and scum behind one of the rotting gates, Riderhood's hold had relaxed, probably in falling, and his eyes were staring upward," [Dickens 1952] which is given a value of –2.15–to the number of strong positive sentiment values in the beginning of chapter 16. Jockers acknowledges the limits of transforming functions in Syuzhet by noting that "when a series of sentence values are combined into a larger chunk using a percentage based measure, extremes of emotional valence tend to get watered down." [Jockers 2017a] The limit of Syuzhet that Jockers admits to does not seem to be applied in isolation to large data, as it is also seen to affect small data. 28

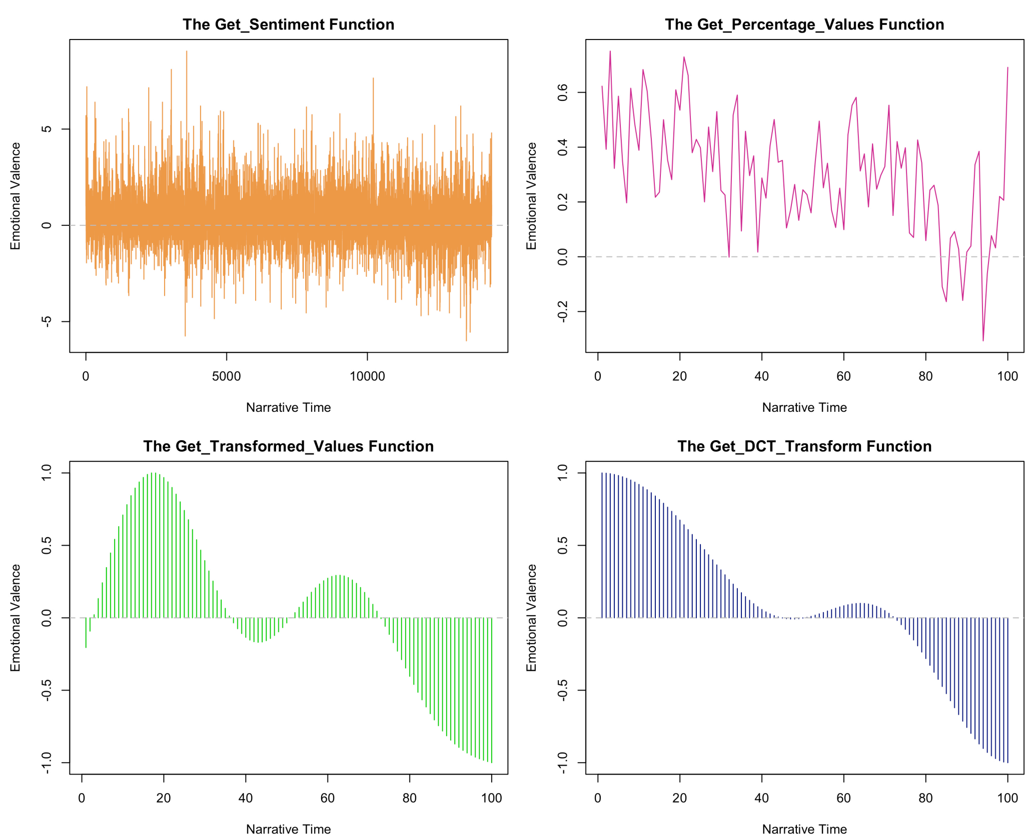

Figure 6. Comparison of four different functions based on the Syuzhet lexicon from George Eliot's *Middlemarch*

Like Dicken's *Our Mutual Friend*, George Eliot's *Middlemarch* is a long Victorian novel, which includes 14,415 sentences after being processed through the get_sentiment function using the Syuzhet lexicon. The number of positive, neutral, and negative sentences 29 from George Eliot's *Middlemarch* was 7,286, 3,017, and 4,112, respectively. The positive and negative averages were 1.09 and –0.88, respectively. The emotional valence from the get_sentiment and the get_percentage_values reveals the dominance of positive emotion throughout the plots, except for the last part, between x≈85 and x≈95. The emotional trajectories from the get_sentiment and the get_percentage_values precisely depict the ambience of its plots. Although *Middlemarch* has a number of conflicts during the course of the novel between Dorothea Brooke and Mr. Casaubon and between Rosamond Vincy and Lydgate, the flow of *Middlemarch* is generally filled with positive feelings with the exception of the end. With the sudden death of Mr. Casaubon and Lydgate, the last part of *Middlemarch* is dominated with negative feelings. However, *Middlemarch* still has a happy ending as Dorothea decides to get married to Will Ladislaw despite the fact that she has to give up her inheritance from Mr. Casaubon when she does so. Rosamond Vincy also remarries another man after losing Lydgate. Mary and Fred live happily together and have children. The happy ending is from x≈98 through 100 (chapter 86 to the finale). The get_sentiment and get_percentage_values functions properly catch the happy ending, whereas the get_transformed_values and get_dct_transform functions do not. In addition, looking into some chapters which have quarrels, there are some parts scored incorrectly by Syuzhet. The highest positive scored sentence is found with a score of 9.05 in chapter 20. Chapter 20 is about the first fight between Dorothea and Mr. Casaubon in Rome after their marriage, which is at x≈25 in figure 6:

> These characteristics, fixed and unchangeable as bone in Mr. Casaubon, might have remained longer unfelt by Dorothea if she had been encouraged to pour forth her girlish and womanly feeling — if he would have held her hands between his and listened with the delight of tenderness and understanding to all the little histories which made up her experience, and would have given her the same sort of intimacy in return, so that the past life of each could be included in their mutual knowledge and affection — or if she could have fed her affection with those childlike caresses which are the bent of every sweet woman, who has begun by showering kisses on the hard pate of her bald doll, creating a happy soul within that woodenness from the wealth of her own love. [Eliot 1967, Chapter 20]

"These characteristics" signifies Mr. Casaubon's "tenacity of occupation and … eagerness." Looking closely into this long sentence, "if" is the key word. Without "if" in this sentence, it would be correct to give this sentence positive scores. In this sentence, there are 20 words which have sentiment scores out of 134 words through the get_tokens and the get_sentiment functions: unchangeable (–0.6), encouraged (0.8), womanly (–0.25), feeling (0.25), delight (1), tenderness (0.8), understanding (1), intimacy (0.8), included (0.6), mutual (0.6), knowledge (0.6), affection (1), affection (1), childlike (0.6), bent (–0.4), sweet (0.75), hard (–0.25), happy (0.75), wealth (0.5), and love (0.75). The sum of the tokens is 10.3, but the sentiment score of the sentence level through the get_sentiment function is 9.05. This is due to the conjunction, "if," which affects the sentence level by adding –0.25 with the get_sentiment function, though it does not have a sentiment score as a word. The word "affection" (1) appeared twice, so "affection" (1) was only added once in the sentence level, which reveals that Syuzhet avoids summing duplicate sentiment words in sentence levels. The logic used by Syuzhet is meticulous in order to differentiate word and sentence levels. However, Syuzhet failed to semantically detect this sentence and created a faulty sentiment result. This long sentence would have been given negative scores if Syuzhet had a function to semantically detect sentences. In addition, there is another example to examine, which is the second highest scored sentence at 8.1 in chapter 16, which is at x≈21 in figure 6: 30

> In Rosamond's romance it was not necessary to imagine much about the inward life of the hero, or of his serious business in the world: of course, he had a profession and was clever, as well as sufficiently handsome; but the piquant fact about Lydgate was his good birth, which distinguished him from all Middlemarch admirers, and presented marriage as a prospect of rising in rank and getting a little nearer to that celestial condition on earth in which she would have nothing to do with vulgar people, and perhaps at last associate with relatives quite equal to the county people who looked down on the Middlemarchers. [Eliot 1967, Chapter 16]

As seen in the passage above, British authors such as George Eliot, Charles Dickens, and Charlotte Brontë intentionally used colons or semicolons to break long sentences into several parts. Since Syuzhet does not split sentences based on colons, the sentences in the passage above were not separated. This passage reveals Rosamond's only reason for caring about Lydgate, which is his social rank. It would be more appropriate to consider this passage as having neutral emotion since it is based on Rosamond's criteria in choosing her husband. In this excerpt, there are 14 words which have sentiment scores out of 108 words through the get_tokens and the get_sentiment functions: romance (0.5), hero (0.75), profession (0.25), clever (0.75), well (0.8), sufficiently (1), handsome (1), good (0.75), birth (0.6), distinguished (0.6), marriage (0.6), prospect (0.6), celestial (0.4), and vulgar (–0.5). There is no duplicates or conjunctions which would make a different sum between the bag of tokens and the bag of sentences. In addition, some words in this part which might have been considered "negative" have not been scored by the Syuzhet lexicon, such as "piquant" and "look down." Syuzhet simply added the sum of sentiment words, and concluded this part to be the second highest positive sentence in *Middlemarch*. 31

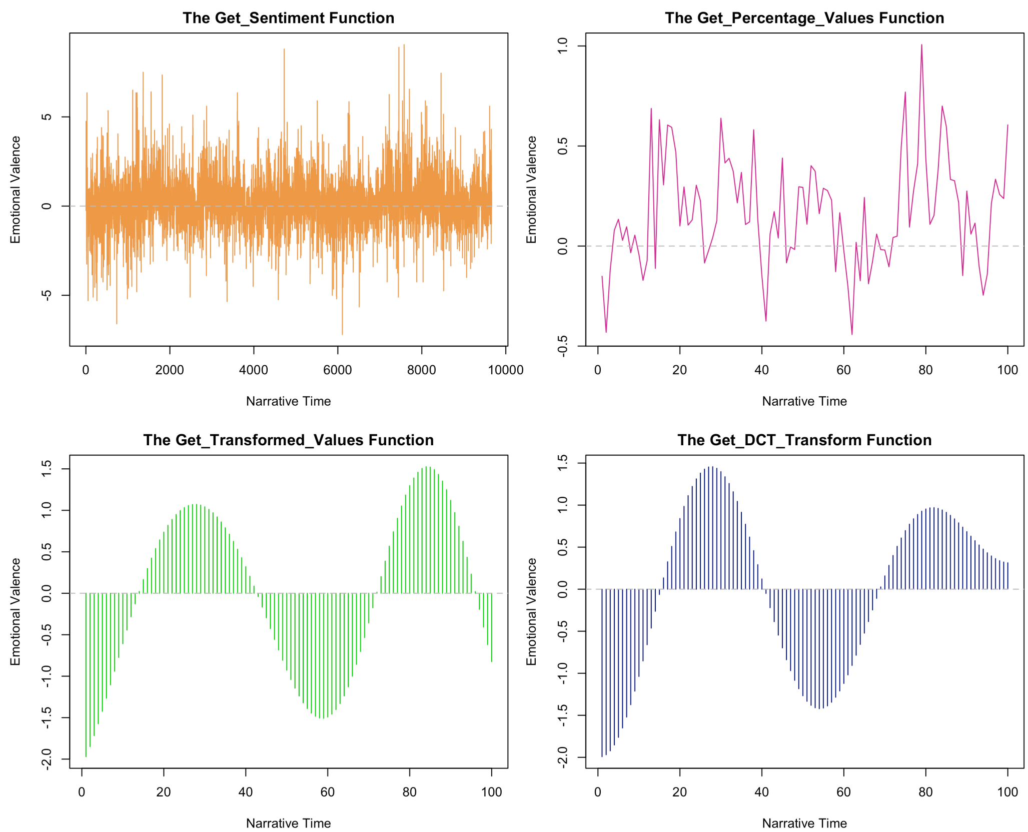

Figure 7. Comparison of four different functions based on the Syuzhet lexicon from Charlotte Brontë's *Jane Eyre*

Charlotte Brontë's *Jane Eyre*, after being processed through the get_sentiment function using the Syuzhet lexicon, included 9,664 sentences. Out of these, 2,776 sentences scored 0 (neutral), 4,046 sentences were positive, and 2,824 sentences were negative. The positive average was 1.08, and the negative average was –0.97. Based on the emotion trajectories created by the four functions, emotions from *Jane Eyre* fluctuate between positive and negative feelings throughout the whole plot. The get_dct_transform result depicts the emotional flow of *Jane Eyre* as fluctuating between negative, positive, negative, and finally positive feelings, whereas the get_percentage_values scrupulously delineates each part with binary emotions. Jane Eyre has difficult times when staying at Gateshead and Lowood due to Mrs. Reed, John Reed, and Mr. Broklehurst, in addition to Helen's death, which occurs from x≈1 to x≈14. Once Jane moves to Thornfield, she has happier days as Adèle's governess with the slow growth of her feelings for Rochester until her wedding. Based on the get_percentage_values function, the flow of the emotional valence is positive between x≈15 and x≈60 except for at x≈41. Chapter 20 is full of negative feelings due to Bertha Mason's attack on Richard Mason, which occurs at x≈41 in figure 7. There is a strong negative spike at x≈41: "I saw Mr. Rochester shudder: a singularly marked expression of disgust, horror, hatred, warped his countenance almost to distortion; but he only said — 'Come, be silent, Richard, and never mind her gibberish: don't repeat it,'"[Brontë 1973] which is given a score of –4.5. After the chapter, the flow of the emotional valence is positive until the wedding day. The get_percentage_values function correctly depicts the emotional valence of this part, whereas the get_dct_transform does not. The wedding was canceled with Mr. Mason's disclosure of the fact that Rochester is already married. Jane reveals her severe feelings when deciding to leave Thornfield: "I wrestled with my own resolution: I wanted to be weak that I might avoid the awful passage of further suffering I saw laid out for me; and Conscience, turned tyrant, held Passion by the throat, told her tauntingly, she had yet but dipped her dainty foot in the slough, and swore that with that arm of iron he would thrust her down to unsounded depths of agony," [Brontë 1973, Chapter 27] which is given a score of –4.65 by Syuzhet at x≈61. After her marriage is canceled, Jane's hardships continue as a street beggar until she settles in at Moor House and Morton. Jane moves to a small cottage, and again experiences a positive life as a teacher at x≈76 (Chapter 31). When Jane finds Rochester in Ferndean, there are sentences which reveal negative emotions: "He [Rochester] was taken out from under the ruins, alive, but sadly hurt: a beam had fallen in such a way as to protect him partly; but one eye was knocked out, and one hand so crushed that Mr. Carter, the surgeon, had to amputate it directly,"[Brontë 1973] which is given a score of –3.25 by Syuzhet at x≈93 (Chapter 36), and which the get_percentage_values function detects precisely. The ending of *Jane Eyre* arouses positive feelings with the successful marriage of Jane and Rochester.

The most negative sentence from *Jane Eyre* has a score of –7.2 in chapter 27, which occurs at x≈63 in figure 7, where Rochester explains about Bertha Mason after the cancellation of their wedding. 33

> These were vile discoveries; but except for the treachery of concealment, I should have made them no subject of reproach to my wife, even when I found her nature wholly alien to mine, her tastes obnoxious to me, her cast of mind common, low, narrow, and singularly incapable of being led to anything higher, expanded to anything larger — when I found that I could not pass a single evening, nor even a single hour of the day with her in comfort; that kindly conversation could not be sustained between us, because whatever topic I started, immediately received from her a turn at once coarse and trite, perverse and imbecile — when I perceived that I should never have a quiet or settled household, because no servant would bear the continued outbreaks of her violent and unreasonable temper, or the vexations of her absurd, contradictory, exacting orders — even then I restrained myself: I eschewed upbraiding, I curtailed remonstrance; I tried to devour my repentance and disgust in secret; I repressed the deep antipathy I felt. [Brontë 1973, Chapter 27]

This passage reveals that Syuzhet does not split sentences based on dashes and semicolons. In this excerpt, there are 27 words which have sentiment scores out of 173 words through the get_tokens and the get_sentiment functions: vile (-0.75), treachery (-0.5), concealment (-0.8), reproach (-0.5), found (0.6), alien (-0.6), obnoxious (-0.75), incapable (-0.75), led (0.4), found (0.6), comfort (0.75), kindly (0.5), received (0.6), coarse (-0.6), perverse (-0.5), imbecile (-0.75), quiet (0.25), household (0.6), violent (-0.75), unreasonable (-0.5), temper (-0.5), absurd (-0.75), contradictory (-0.5), exacting (-0.25), devour (-0.4), disgust (-1), and antipathy (-0.5). The sum of the word tokens is -7.35. After excluding the duplicated word, "found," the sum should be -7.95, but the Syuzhet score is -7.2. This is because Syuzhet perceives words with dashes as being together. In this part, "imbecile" should have been counted as -0.75, but "imbecile" was processed as "imbecile — when," which is considered null by Syuzhet. Although Syuzhet successfully labeled this part as negative, it shows the limits of the Syuzhet functions. 34

The most positive sentence from *Jane Eyre* scored a 9.05 in chapter 32, which is at x≈78 in figure 7: 35

> She was hasty, but good-humoured; vain (she could not help it, when every glance in the glass showed her such a flush of loveliness), but not affected; liberal-handed; innocent of the pride of wealth; ingenuous; sufficiently intelligent; gay, lively, and unthinking: she was very charming, in short, even to a cool observer of her own sex like me; but she was not profoundly interesting or thoroughly impressive [Brontë 1973, Chapter 32]

This is Jane's positive description of Rosamond Oliver. In this part, there are 19 words which have sentiment scores out of 69 words through the get_tokens and the get_sentiment functions: hasty (–0.5), good (0.75), vain (–1), flush (–0.4), loveliness (1), innocent (0.8), pride (0.25), wealth (0.5), ingenuous (1), sufficiently (1), intelligent (1), lively (0.75), charming (1), cool (0.75), sex (0.1), like (0.5), profoundly (0.8), interesting (0.75), and impressive (0.75). Syuzhet seems to successfully detect this part as positive. The original score should be 9.8 instead of 9.05 since "good-humoured" was not separately detected in the sentence level due to the dash, which means "good" (0.75) was not counted towards the sentiment score sum in this part. However, in the last sentence, "but she was not profoundly interesting or thoroughly impressive," Syuzhet failed to detect the negation "not" and simply added scores from the words, profoundly (0.8), interesting (0.75), and impressive (0.75) without reversing them, which brought about incorrect results. 36

5. Conclusion

Through the sentiment analysis of the three novels, the get_transformed_values and get_dct_transform functions do not indicate sophistication of emotion, since their purpose is to grasp the whole emotional flow of plots by simplifying the emotional valence with a smoothing filter, whereas the get_sentiment and get_percentage_values functions create more detailed results of the emotional valence, which is more appropriate for micro sentiment analysis. Syuzhet reveals its limits through the lack of functions to detect dashes, negators, and adversative conjunctions/transitions, which brings about faulty results. Syuzhet does not detect the syntactical and semantic information of each sentence, but simply transforms each word found in the lexicons into numerical sentiment vectors. In addition, the application of DCT for sentiment analysis of literary texts is still questionable as the graphs of sentiment analysis generated with DCT are over-simplified and often incorrect. Syuzhet has been the most popular sentiment analysis tool for R despite its limits. However, it will continue to be questionable as a research tool in the digital humanities without overcoming the limits mentioned above. 37

Sentiment analysis has been developed with the implementation of machine learning and deep learning approaches, which attempt to solve the issues it faces. Deep learning in natural language process has shown a shining future for sentiment analysis. For 38 example, convolutional neural networks (CNN) which include the convolution stage, detector stage, and pooling stage can improve the accuracy of sentiment analysis by detecting locality and negativity of words. Bing Liu notes that opinion words have different meanings depending on the context [Liu 2010, 16]. For example, the sentences, “I am not happy to work out” and “I am happy not to work out,” have different meanings. The locality of “not” can be processed in pooling layers, which are usually applied after the convolutional and detector stages. For example, MALLET (MAchine Learning for LanguagE Toolkit), [12] a text mining toolkit, employs conditional random fields (CRF), including the Naïve Bayes classifier and decision trees. CRF is an efficient method of natural language processing that fixes the issues of two previous models, namely HMM [Hidden Markov Model] and MEMM [Maximum Entropy Markov Model]. Using deep neural network (DNN) models with word embeddings, which are “typically pre-trained,” made it possible for the learned word vectors to “capture general syntactical and semantic information.” [Do et al. 2019, 276] Similarly, the BERT model, which was created and released by Google AI researchers in 2018, possesses the possibility of application for sentiment analysis with literary texts. BERT, a bidirectional language model, performs a variety of natural language tasks based on a pretrained model with a deep bidirectional transformer that achieves “state-of-the-art performance on a large suite of sentence-level *and* token-level tasks.” [Devlin et al. 2019, 4172] The recent experiment by Haider et al. [Haider et al. 2020] revealed the inconsistent results of fine-tuning the BERT-Base model for the sentiment analysis of poems, due to the lack of vocabulary in poems. While deep learning cannot achieve perfect results, current research shows that deep-learning based sentiment analysis has higher accuracy than lexicon-based sentiment analysis. Stephen Ramsay mentions that “the real failure would not be a result that is deemed incorrect” but “the decision to banish” computational literary analysis entirely [Ramsay 2016, 529]. Although it would require the collaborative creation of literary datasets for deep learning-based sentiment analysis, we should strive to implement deep learning models for sentiment analysis in the digital humanities.

It is painstaking to improve the precision, accuracy, and efficiency of digital tools, and the process entails a great deal of effort, emotion, time, and money, which are also needed to maintain tools after development. Some scholars show disdain for and misunderstanding of the funding necessities for DH projects by stating that “almost all of the works” can be recreated with only one laptop [Da 2019, 603] As a mobile/web developer, whenever I had meetings with clients interested in making apps without in-depth knowledge in the IT field, there was always a common qualm about costs to develop apps, before they even thought about the cost of future maintenance. To create a simple app that contains only a few functions requires a project manager, iOS/Android developers, back-end developers, and an UI/UX designer. DH projects are no different: Amy Earhart and Toniesha Taylor shared their experiences facing institutional funding issues while collaborating on a DH project [Earhart and Taylor 2016]. Due to insufficient funding in the humanities field, it will be challenging to develop new algorithms/functions and maintain Syuzhet. Syuzhet is a free digital tool that will continue to be developed even though, like any other existing computer program, it is not perfect. I believe that the necessary improvements will be made to Syuzhet for semantically and syntactically detecting sentences, so long as digital humanists support Syuzhet. Improving sentiment analysis as well as digital tools should not remain only as the duty of developers or labs, but as a responsibility of all digital humanists who employ digital tools by participating in making improvements through the provision of feedback, such as that in Swafford’s blog post. We need to keep testing and providing feedback to improve tools like Syuzhet for the affluence, development, and application of sentiment analysis in literature. 39

Acknowledgements

I would like to express my appreciation to Amy Earhart, Ruihong Huang, Maura Ives, Shawna Ross, Lindsey Kim, Shawna Campbell, Quinn Dombroski, and Kyle Fisher for their feedback, suggestions, and supoprt. In addition, I thank the editors and reviewers for their time and constructive feedback. 40

Notes

[1] Laura Mandell expressed her qualms about sentiment analysis after reading Swafford’s post, “Problems with the Syuzhet Package.” [Swafford 2015] Jonathan Goodwin shared the incorrect results of Syuzhet on Twitter (See https://twitter.com/joncgoodwin/status/563734388484354048/photo/1).

[2] Check the update notes for Syuzhet at https://github.com/mjockers/syuzhet/blob/master/NEWS. Although some versions of Syuzhet, including Syuzhet 1.0.5, were annotated in the note, some of them were not released to the public.

[3] VADER is also a popular sentiment analysis tool in Python, but the number of VADER downloads in R is low since it was only recently released, on May 22, 2020.

[4] The Bing lexicon was created by Bing Liu and collaborators.

[5] The Afinn lexicon was created by Finn Årup Nielsen.

[6] The NRC lexicon was created by Saif M. Mohammad and Peter D. Turney.

[7] See descriptions of inquirer categories and use of inquirer dictionaries: https://www.wjh.harvard.edu/~inquirer/homecat.htm.

[8] See Apache OpenNLP developer documentation: https://opennlp.apache.org/docs/1.9.1/manual/opennlp.html.

[9] See Stanford parser: http://nlp.stanford.edu:8080/parser/index.jsp.

[10] I excluded the test results of Henry's finance-specific dictionary (HE) since they are all zero.

[11] The sentence "She was happy but she is no longer happy" created different sentiment values between the VADER R and Python packages, with –0.391 in VADER 0.2.1 in R released on September 7, 2020 and –0.665 in VADER 3.3.2 in Python released on July 27, 2018, respectively.

[12] See MAchine Learning for LanguagE Toolkit: http://mallet.cs.umass.edu.

Works Cited

Binder 2016 Binder, J. M. (2016). "Alien Reading: Text Mining, Language Standardization, and the Humanities." *Debates in the Digital Humanities 2016,* pp. 201–17.

Brontë 1973 Brontë, C. (1973). *Jane Eyre*. Introduction by Margaret Smith. London: Oxford University Press. (Originally published 1847)

Clement 2016 Clement, T. E. (2016). "The Ground Truth of DH Text Mining." *Debates in the Digital Humanities 2016*, pp. 534–5.

Da 2019 Da, N. Z. (2019). "The Computational Case against Computational Literary Studies." *Critical Inquiry,* 45(3), pp. 601–39. https://doi.org/10.1086/702594.

Devlin et al. 2019 Devlin, J., Chang, M. W., Lee, K., and Toutanova, K. (2019). "BERT: Pre-training of deep bidirectional transformers for language understanding." In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, vol. 1, pp. 4171–86.

Dickens 1948 Dickens, C. (1948). *Bleak House*. Introduction by Sir Osbert Sitwell. London: Oxford University Press. (Originally serialized 1852–53)

Dickens 1952 Dickens, C. (1952). *Our Mutual Friend*. Introduction by Salter Davies. London: Oxford University Press. (Originally serialized 1864–65)

Do et al. 2019 Do, H. H., Prasad P. W. C., Maag A. and Alsadoon A. (2019). "Deep Learning for Aspect-Based Sentiment Analysis: A Comparative Review." *Expert Systems with Applications*, 118: pp. 272–99. https://doi.org/10.1016/j.eswa.2018.10.003.

Earhart and Taylor 2016 Earhart, A. E. and Taylor, T. L. (2016). "Pedagogies of Race: Digital Humanities in the Age of Ferguson." *Debates in the Digital Humanities 2016*, pp. 251–64.

Eliot 1967 Eliot, G. (1967). *Middlemarch*. Introduction by Barbara Hardy. London: Oxford University Press. (Originally serialized 1871–72)

Habimana et al. 2020 Habimana, O., Li, Y., Li, R., Gu, X., and Yu, G. (2020). "Sentiment Analysis Using Deep Learning Approaches: An Overview." *Science China Information Sciences*, 63(1), pp. 1–36.

Haider et al. 2020 Haider, T., Eger, S., Kim, E., Klinger, R., and Menninghaus, W. (2020). "PO-EMO: Conceptualization, Annotation, and Modeling of Aesthetic Emotions in German and English Poetry." *International Conference on Language Resources and Evaluation (LREC) 2020*, Marseille.

Jockers 2015 Jockers, M. L. (2015). Blog comment on "Problems with the Syuzhet Package." *Anglophile in Academia: Annie Swafford's Blog.* https://annieswafford.wordpress.com/2015/03/02/syuzhet/comment-page-1/#comment-54.

Jockers 2017a Jockers, M. L. (2017). "Introduction to the Syuzhet Package." https://cran.r-project.org/web/packages/syuzhet/vignettes/syuzhet-vignette.html.

Jockers 2017b Jockers, M. L. (2017). "Syuzhet Package v1.0.4." *R Documentation.* https://www.rdocumentation.org/packages/syuzhet/versions/1.0.4.

Jung et al. 2008 Jung, Y., Choi, Y. and Myaeng, S. (2008). "A Study on Negation Handling and Term Weighting Schemes and Their Effects on Mood-Based Text Classification." *Korean Journal of Cognitive Science*, 19: pp. 477–97.

Kim and Klinger 2018 Kim, E., and Klinger, R. (2018). "A Survey on Sentiment and Emotion Analysis for Computational Literary Studies." *arXiv* preprint. *arXiv*:1808.03137.

Kwartler 2017 Kwartler, T. (2017). *Text Mining in Practice with R*. John Wiley and Sons.

Liu 2010 Liu, B. (2010). "Sentiment Analysis and Subjectivity." *Handbook of Natural Language Processing*, 2: pp. 627–66. https://www.cs.uic.edu/~liub/FBS/NLP-handbook-sentiment-analysis.pdf.

Medhat et al. 2014 Medhat, W., Hassan, A. and Korashy, H. (2014). "Sentiment Analysis Algorithms and Applications: A Survey." *Ain*

Shams Engineering Journal 5(4): pp. 1093–113. https://doi.org/10.1016/j.asej.2014.04.011.

Misuraca et al. 2020 Misuraca, M., Forciniti, A., Scepi, G., and Spano, M. (2020). "Sentiment Analysis for Education with R: Packages, Methods and Practical Applications." *arXiv*:2005.12840.

Mohammad 2012 Mohammad, Saif M. (2012). "From Once Upon a Time to Happily Ever After: Tracking Emotions in Mail and Books." *Decision Support Systems*, 53(4), pp. 730–41.

Ramsay 2011 Ramsay, S. (2011). *Reading Machines: Toward an Algorithmic Criticism*. Champaign: University of Illinois Press.

Ramsay 2016 Ramsay, S. (2016). "Humane Computation." *Debates in the Digital Humanities 2016*, pp. 527–9.

Reagan et al. 2016 Reagan, A. J., Mitchell, L., Kiley, D., Danforth, C. M., and Dodds, P. S. (2016). "The Emotional Arcs of Stories are Dominated by Six Basic Shapes." *EPJ Data Science*, 5(1), pp. 1–12.

Rhody 2015 Rhody, L. (2015). Blog comment on "Problems with the Syuzhet Package." *Anglophile in Academia: Annie Swafford's Blog.* https://annieswafford.wordpress.com/2015/03/02/syuzhet/comment-page-1/#comment-59.

Sinclair et al. 2013 Sinclair, S., Ruecker, S. and Radzikowska, M. (2013). "Information Visualization for Humanities Scholars." *Literary Studies in the Digital Age-An Evolving Anthology*. DOI: 10.1632/lsda.2013.6.

Swafford 2015 Swafford, A. (2015). "Problems with the Syuzhet Package." *Anglophile in Academia: Annie Swafford's Blog.* https://annieswafford.wordpress.com/2015/03/02/syuzhet/.

Underwood 2014 Underwood, T. (2014). "Theorizing Research Practices We Forgot to Theorize Twenty Years Ago." *Representations*, 127(1): 64–72.

Welch and Mihalcea 2016 Welch, C., and Mihalcea, R. (2016). "Targeted Sentiment to Understand Student Comments." *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*: pp. 2471–81.