[preprint]

Sentiment Analysis of Linguistic Data in Behavioral Research

Ian Cero[1], Jiebo Luo[2], & John Michael Falligant[3,4]
May 23, 2022

Author Affiliation

1. Department of Psychiatry, University of Rochester Medical Center, Rochester, NY, USA
2. Department of Computer Science, University of Rochester, Rochester, NY, USA
3. Department of Psychiatry and Behavioral Sciences, Johns Hopkins University School of Medicine, Baltimore, MD, USA
4. Department of Behavioral Psychology, Kennedy Krieger Institute, Baltimore, MD, USA

Author Note

Correspondence concerning this article should be address to Ian Cero, University of Rochester Medical Center, 300 Crittenden Blvd, Rochester, NY 14642. Email: ian_cero@urmc.rochester.edu

Statements and Declarations

The authors declare that they have no financial or non-financial interests that are directly or indirectly related to the work submitted for publication

Acknowledgments

This work was supported by a grant (KL2 TR001999) from National Center for Advancing Translational Sciences (NCATS) at the National Institutes of Health (NIH). It was also supported by a National Institutes of Health Extramural Loan Repayment Award for Clinical Research (L30 MH120727).

Orcid Information

- Ian Cero: 0000-0002-2862-0450
- Jiebo Luo: 0000-0002-4516-9729
- John Michael Falligant: 0000-0002-0062-9074

Abstract

A complete science of human behavior requires a comprehensive account of the verbal behavior those humans exhibit. Existing behavioral theories of such verbal behavior have produced compelling insight into language's underlying function, but the expansive program of research those theories deserve has unfortunately been slow to develop. We argue that the status quo's manually implemented and study-specific coding systems are too resource intensive to be worthwhile for most behavior analysts. These high input costs in turn discourage research on verbal behavior overall. We propose lexicon-based sentiment analysis as a more modern and efficient approach to the study of human verbal products, especially naturally-occurring ones (e.g., psychotherapy transcripts, social media posts). In the present discussion, we introduce the reader to principles of sentiment analysis, highlighting its usefulness as a behavior analytic tool for the study of verbal behavior. We conclude with an outline of approaches for handling some of the more complex forms of speech, like negation, sarcasm, and speculation. The appendix also provides a worked example of how sentiment analysis could be applied to existing questions in behavior analysis, complete with code that readers can incorporate into their own work.

Keywords: data science; matching; natural language processing; sentiment; text analysis; verbal behavior

Sentiment Analysis of Linguistic Data in Behavioral Research

Most human affairs are verbally mediated. A comprehensive account of human verbal behavior is thus necessary for a truly complete science of human behavior. Although existing theories have made substantial progress toward this goal (Hayes et al., 2001; Skinner, 1957), the study of naturally-occuring verbal behavior remains limited by at least two practical barriers. First, time-intensive hand coding is often required to wrangle raw verbal data into an analyzable format. Second, the sheer complexity of a real-world verbal scenario often requires a bespoke coding scheme for each study. This lack of methodological standardization in turn leads to duplication of work across different studies and complicates evaluation of the verbal behavioral research literature for its consumers (see Critchfield, Becirevic, et al., 2017). The lack of a coherent methodology for examining verbally mediated phenomena may also reflect broader issues within the discipline of behavior analysis with respect to communication of practices and findings with other professionals and consumers (see Becirevic et al., 2016; Critchfield et al., 2016; Critchfield & Doepke, 2018; Critchfield, Doepke, et al., 2017).

Fortunately recent developments in Computer and Data Science have greatly improved the prospects for behavior scientists studying verbal behavior, especially naturally-occuring verbal behavior. For example, Natural Language Processing has produced a range of impressive techniques for describing linguistic products and using them to predict aspects of human behavior (Jurafsky & Martin, 2008), including in mental health (De Choudhury et al., 2014, 2016). The most relevant of these techniques for contemporary behavior scientists is Sentiment Analysis, the subfield of Natural Language Processing focused on measuring the overall ‘sentiment’ of a verbal product (e.g., session transcripts, diary entries, social media posts; Liu, 2020). Most often, the sentiment being estimated is the general positivity or negativity of a verbal product. However, with minor adjustments, the same techniques can be

used to quantify a range of emotions, as well as social and intellectual processes (Tausczik & Pennebaker, 2010).

This discussion is thus designed to introduce readers to the theory and practice of Sentiment Analysis, with an emphasis on how those techniques can be applied to the study of verbal behavior. We start by outlining some unsolved practical barriers that arise from the status quo approach for studying verbal products. We then introduce readers to the assumptions and techniques of sentiment analysis. Lastly, we address some of the most common objections to sentiment analysis and outline potential solutions to them. An additional worked example is also given in the appendix.

Status Quo Stagnation

Existing theories of verbal behavior have generated substantial insight into not only the underlying taxonomy of verbal products humans exhibit, but also the functional relations that causally connect those products to stimuli in the surrounding environment (Hayes et al., 2001; Skinner, 1957). A nuanced account of verbal operants is required to understand and change complex, socially significant human behavior. There have been many calls for behavior analysts to broaden the scope of their research and clinical practice to incorporate dimensions of human experience (e.g., emotion, linguistics) often untouched by behavior analysts. As described by Hayes (2001), a comprehensive experimental behavior-analytic account of psychology requires a continuous revitalization of behavior-analytic methodologies, and an openness to new theories, preparations, and concepts. But as the giants of these same theories have themselves argued (Hayes et al., 2001), an expansive program of research into naturally-occuring verbal behavior has still been slow to develop. Even giving ample credit to novel laboratory tools like the Implicit Relational Association Test (Barnes-Holmes et al., 2008; Hussey et al., 2015) and the Function Acquisition Speed Test (O'Reilly et al., 2012), research on the naturally occurring verbal products characterizing most humans' day-to-day experience (e.g., conversation transcripts, social media posts) has largely failed to emerge (cf. Critchfield & Doepke, 2018;

Reed, 2016). The lack of systematic research on these verbal relations also has direct implications for the dissemination of behavior-analytic research, as some have highlighted the iatrogenic potential of behavior-analytic jargon on perceptions of our methodologies and applied practices (Becirevic et al., 2016; Critchfield et al., 2016; Critchfield & Doepke, 2018; Critchfield et al., 2017; Normand & Donohue, 2022)

A comprehensive analysis of the growth rate in the study of verbal behavior is unfortunately beyond the scope of this study. But to motivate the discussion, we highlight one of our favorite papers as an unfortunate case in point. Recall McDowell and Caron's (2010) finding that the amount of rule-breaking talk produced by potentially delinquent boys was directly proportional to the amount of positive peer talk that followed those utterances. Even more interesting, the association between rule-break talk and positive peer talk strongly conformed to the a priori predictions of the general matching law (GML; Herrnstein, 1970; McDowell, 2013), explaining an average 90% of the variance in the amount of rule-break talk produced. In fact, the bias parameter (*b*) of the GML for rule-break talk scaled to a measure of deviancy, illustrating that the verbal products of conversation directly map onto real-world measures of target behavior. This study has many upsides. It is theory-driven, has applied relevance across the social and behavioral sciences, and contains clear implications for a future program of experimental and applied interventional studies (e.g., Luna, 2019; Simon & Baum, 2017). But among its 35 citations at the time of this writing, a vast majority are reviews, position papers, or commentaries. It appears to have generated conceptual interest, but little empirical follow up. Where is the subsequent program of empirical research this paper deserves?

We argue this paper shares an important feature with many others investigating naturally-occuring verbal behavior (e.g., Critchfield et al., 2016), one that makes it especially hard to replicate and extend. Specifically, it uses a *bespoke manual coding* (BMC), in which a custom coding scheme is developed for this (and only this) study and then time-intensively

implemented by trained observers. This approach has two major drawbacks, which are no longer necessary in most research contexts.

First, although the BMC approach is easy to conceive, it is resource intensive to implement. In the study above, each of the 210 subject pairs produced 20 minutes of speech that needed to be hand-coded multiple times by different observers. Assuming only two observers that could code a transcript at the same rate as it would take to read a transcript out loud, that yields 2 × 20 × 210 / 60 = 140 person hours just to complete the coding. This is a big investment just to go from raw text to spreadsheet (not to mention the time and effort quantifying the degree of interobserver agreement between coders and resolving discrepancies when they arise). Replicating and extending this important paper thus comes at a relatively high cost, in turn reducing the likelihood it ever replicated or extended.

Second, the BMC approach is typically designed to be implemented in a specific study and then discarded. Although this increases the likelihood that a given coding scheme will be sensitive to the unique features of a given study population, it slows the incremental progress that could have been produced by a collection of studies (and their tools) that build on one another. Although this collective goods problem is not the fault of any particular researcher or study, the BMC norm has preserved precision at the cost of progress. In response, we argue sentiment analysis is a better trade for behavior analysts studying verbal behavior, especially in naturally-occuring contexts. As we show below, it is (a) orders of magnitude more efficient to implement than the BMC approach. Moverover, (b) the use of a standardized coding scheme allows an incremental consensus to develop over time, including even a consensus that we should discard that specific scheme for a new one.

Sentiment Analysis

The goal of Sentiment Analysis is to quantify the tone or “sentiment” of verbal products. Most often, the sentiment of greatest interest is a document’s *polarity* or *valence*, which reflects overall how positive or negative the views expressed in that document are. In addition to the

valence of a document, Sentiment Analysis can also be used to quantify more specific emotional features, like sadness or depression (De Choudhury et al., 2014; Dodds et al., 2011). With slightly more effort, research relying on sentiment analysis techniques has also attempted to quantify even more complex social and intellectual reasoning processes (Tausczik & Pennebaker, 2010).

How can we quantify the valence, emotion, or even reasoning processes expressed in a document? In practice, there are two main approaches used to achieve this task: *lexicon-based* and *machine learning-based*. Although head-to-head comparisons show that learning-based methods are often the more accurate of the two approaches (Kotelnikova et al., 2021; H. Zhang et al., 2014), lexicon-based approaches remain popular and competitive because they are easier to implement and interpret. Lexicon-based methods are also more effective in situations with smaller samples and when there is a focus on specific parts of a document (e.g., praise directed to a specific learner), rather than the overall tone of that document (Khoo & Johnkhan, 2018). Overall, the lexicon-based approach will be more practical, adequate, and insight generating for most behavior analysts than the learning-based approach. We therefore focus exclusively on lexicon-based approaches in this discussion, referring interested readers to more comprehensive works for an introduction to machine learning-based strategies (e.g., Liu, 2020) and their recent application to behavior analysis (Bailey et al., 2021; Lanovaz et al., 2020; Lanovaz & Hranchuk, 2021; Taylor & Lanovaz, 2021; Turgeon & Lanovaz, 2020).

Background Assumptions

Lexicon-based sentiment analysis rests on the twin assumptions that (a) words and other tokens have at least some semantic orientation that they retain across contexts and that (b) these semi-stable orientations can be used to infer the approximate sentiment of a given document, despite the unique context in which the document was created. Phrased more practically, word meanings might fluctuate from context to context, but they have a kind of typical, consensus, or average meaning around which they orbit. Thus, by simply guessing that

this typical meaning of the word holds in a given document, we can expect to be approximately right much of the time.[1]

To see this context-free semantic orientation in action, note English-fluent readers will accurately sense “murder” has a negative polarity that persists across most contexts. Even overall positive phrases like, “I was ultimately glad he was murdered” imply that the typical reaction to murder is negative. This same assumption extends to larger units of speech too. “Not happy” has at least some context-free negative valence that persists across the different ways that phrase can be used. Those two words, placed in that order, will tend to imply something negative. The same assumption also extends to word qualities beyond just positive-negative polarity. Words like “bonus” almost always imply some degree of uncertainty or surprise. Likewise, it is also plausible many other words have some context-free semantic orientations representing even more complex processes. For example, “because” and “effect” are related to causal processes and therefore signal at least something about the kind of intellectual orientation in which a speaker was engaged.

If our anecdotal experience with colleagues holds, most readers can immediately think of several ways these assumptions can go awry. The most obvious case is sarcasm, in which a speaker expresses literal agreement or support, but a combination of tone and context indicate their true intention was to communicate disagreement or criticism. These readers have correctly intuited that such creative uses of language are inconsistent with the background assumptions of lexicon-based sentiment analysis and thus the method is often fooled by them. To ensure concerned readers do not immediately abandon us here, please note that we address this and other complex cases (e.g., negation, speculation) directly in the final section - after readers are more familiar with sentiment analysis as a whole. We also note that sarcasm detection even by

[1] Whether these key assumptions hold well enough for a given application is of course an empirical question. However, some version of them must be true in order for humans to communicate at all. For if the meanings of words were totally unique from context to context, verbal communication itself would be impossible.

machine learning models also remains a challenging open problem at present (Joshi et al., 2016).

Conducting sentiment analysis

In sentiment analysis, verbal products can be thought of as *documents* that contain *tokens*. Here, we use the term *document* to refer to any verbal product that could in principle be converted into text. Watercooler conversations, phone calls, and social media posts can all be thought of as "documents" in this sense. In contrast, tokens are the smallest units over verbal behavior under consideration and can be anything from individual letters, to words, to clauses, to sentences, and so on. Individual words or small collections of adjacent words called *n-grams* are by far the most common. For example, the word "happy" is a *unigram*, "not happy" is a *bigram*, "am not happy" is a *trigram*, and so on. To analyze the verbal content of a document, it must typically be broken up into individual tokens. Documents that are broken up into a list of tokens are thus said to have been *tokenized*. Sometimes, different forms of the same token are further standardized with a process called stemming (removing frills like suffixes) or lemmatization (a more complex approach for finding the root or *lemma* of a word), but that will be unnecessary for the approach we take here.

Lastly, a *lexicon* is a dictionary that links tokens - again typically words, or collections of a few words - to the (hopefully) context-independent categories or numerical values they represent. This is illustrated in Table 1, which contains example entries from three popular lexicons (Hu & Liu, 2004; Mohammad & Turney, 2013b; Nielsen, 2011). Importantly, not all lexicons cover the same collection of words, resulting in some missing cells in the table. So it is important to consider whether a lexicon covers the kinds of tokens likely to be used in a population or context. For example, Western Internet slang often uses non-alphabetic emoticons or emojis (e.g., ":)", ">:(") to communicate important aspects of tone and research studies that are likely to encounter many tokens of that kind will benefit from lexicons that include them (e.g., Dodds et al., 2011). Moreover, not all lexicons categorize words in the same way. For example,

the NRC lexicon provides 8 potential emotions a word might be associated with, along with whether that word has a broadly positive or negative association. In contrast, the Bing lexicon categorizes words only by whether they are positive or negative, which is easier to interpret, but is less flexible. Different still, the AFINN lexicon provides integer ratings for only the positiveness or negativeness of a word, rated from -5 to +5.

The process of conducting a sentiment analysis thus involves applying the background assumptions outlined above to a collection of documents in a systematic way - one that intuitively resembles much of the BMC approach, but sped up with the aid of a computer. These steps are illustrated via an annotated example in Appendix A and involve first selecting a pre-validated lexicon (or creating a new one). Once the lexicon is selected, each document is tokenized into chunks that match the chosen lexicon (e.g., a lexicon designed for bigrams implies that each document should be tokenized into bigrams too). To remove noise from the analysis, the analysis will exclude any *stop words / stop tokens*, which are words or phrases that occur so frequently that they are uninformative about the text from which they are drawn (e.g., “and”, “of the”). The remaining tokens are then scored according to the values present in the lexicon and aggregated (e.g., mean positivity per social media post). This results in a familiar dataset with cases (e.g., subjects, blocks of trials) represented in rows and sentiment scores for each of those cases represented in columns. From there, standard analytic methods can now be straightforwardly applied.

For example, after computing sentiments for several thousand social media posts, the example in Appendix A uses linear regression to evaluate how well the authors of those posts conform to the generalized matching law. That walkthrough also demonstrates several important features of sentiment analysis, including its feasibility for behavioral researchers and its fit with the existing pantheon of methods in behavior analysis. Specifically, in only about 100 lines of human-readable code, we were able to collect 18,000 instances of verbal behavior from two research subjects, score them for their use of previously validated set of trust-related words, and

then show that the use of those words by each subject occurred in proportion to the rate at which it received putative reinforcement (likes from other users). As additional evidence of feasibility, the first author used a time-tracking application to record exactly how many human hours this analysis took to code and run. Results showed the entire code construction process took 4.82 hours from start to finish. Admittedly, the lead author already had experience with this and related techniques, making this a somewhat optimistic estimate for initiates. However, it does demonstrate that with practice, the study of naturally-occuring verbal behavior via sentiment analysis can be far more efficient than the existing BMC approach.

Methods to Address Known Limitations

For the remainder of our discussion, we address the lingering concern about handling complex forms of speech like sarcasm, which we have saved until now so that the reader had the benefit of already seeing a full sentiment analysis for context.

Negation, Sarcasm, and Speculation

Attentive readers will note that the assumptions underlying sentiment analysis - that words have at least *some* context-independent semantic orientations that we can estimate - are more plausible for some styles of naturally-occurring speech than others. For example, “I’m happy” and “I’m not happy” would mistakenly receive the same positive score with the basic techniques we applied above, even though the second speaker is clearly negating the fact that they are currently happy. Complicating things even more, the phrase “I’m *soooo* happy today” could imply extra happiness in the event of sincerity or substantial unhappiness in the event of sarcasm. The basic sentiment analysis techniques we have introduced here would likely fail this test. Equally difficult is the case of speculation (e.g., “I wonder if I will be happy today”). These styles of communication are similar both in that they are quite common and that they modify the usual meaning of the words in our standard lexicons.

Knowing such problems are likely to arise in advance, how can we compute accurate sentiment scores in spite of them? At the outset, we are disappointed to tell the reader that

handling subtle forms of speech is still an open question in sentiment analysis (Joshi et al., 2016; c.f. Dragut & Fellbaum, 2014; Dragut et al., 2014; Emerson & Declerck, 2014; Jia et al., 2009; Schneider & Dragut, 2015). It is a genuinely hard problem and no one solution has produced universal consensus.

Strategy 1: Choose an Honest Operational Definition

One option is to honestly and transparently expand your operational definition of the sentiment you are trying to study, as we have done in our walkthrough. Note, we were careful to say we were analyzing “trust-related” language or “words with a trust dimension,” rather than “words expressing trust.” Our phrasing focuses on the general category to which a word belongs, and is likely expansive enough to safely include cases of negation, sarcasm, and speculation. In contrast, assuming communications that use trust-related words are reliably “expressions of trust” might be more specific, but is also clearly wrong. This careful choice of operational definition is essential for all researchers employing sentiment analysis and thus our recommendation for a minimum standard.

Strategy 2: Hand-Score a Subsample of Verbal Behavior and Compare to Machine-Scored Results

The second approach is to hand-check a random subset of documents that were scored by the machine. This is analogous to existing inter-observer agreement procedures, and will help quantify whether the results of the sentiment analysis are sufficiently accurate for drawing scientifically meaningful conclusions. For example, in a previous study interested in suicide-related words (Cero & Witte, 2020), we asked blinded raters to examine a random subset of tweets that had been machine coded as related or not related to suicide. Results showed raters perceived the authors of putatively suicide-related tweets to be at significantly greater risk of a suicide attempt than the authors of non-suicide-related tweets. Although the scoring procedure was shown to be imperfect, this follow-up check was sufficient to support the remaining claims of the paper - in this case that people talking about suicide do so in social

clusters. Whenever possible, we also recommend this approach for the same reason researchers already evaluate the performance of human-rated coding schemes. It helps quantify the trustworthiness of the results, often revealing that imperfect instruments are reliable enough to safely draw important scientific conclusions.

Strategy 3: Use a More-Complex Lexicon

The third approach is to perform a more complex kind of sentiment analysis that adjusts the score of a given word “happy,” depending on the words around it. This can most easily be done with a lexicon explicitly built for such a purpose. For example, the Sentiment Composition Lexicon of Opposing Polarity Phrases (SCL-OPP) includes collections of emotionally-salient words (e.g., “happy”) that have been paired with negators (“not happy”), modals (“would have been happy”), and degree adverbs (“somewhat happy”) and re-rated for positive / negative polarity (Kiritchenko & Mohammad, 2017). Because this lexicon uses multi word tokens that have already been adjusted, a researcher can apply them using essentially the same code we offer above. However, it is important to remember that there are many more combinations of words than there are individual words. For this reason, multiword lexicons will tend to cover less of the overall space of possible speech a researcher might be interested in studying. For greater accuracy, this reduced coverage will likely be worth the trade, but it is worth knowing about it in advance.

Strategy 4: Use an Adjustment Formula to Re-Weight Negated Words

Another approach has been to try to mathematically adjust the sentiment of a word that has been negated. If happy = +.75, then not happy = -.75. Although intuitive, this approach is ultimately still quite coarse. For example, in contemporary English negating a positive word (“not good”) tends to imply negative sentiment, but negating a negative word (“not bad”) is more likely to imply neutrality than positivity (Hii, 2019; Kiritchenko & Mohammad, 2017). An additional problem comes from *negation scoping* (e.g., Pröllochs et al., 2015). For example, the scope of the negational phrase “I have not been happy in 10 years” is just one or two words preceding

happy and the negation is easy to detect. But in the rearranged case of “happiness is an emotion I have not experienced in 10 years”, the negation follows happiness and is much farther away. How can we detect the scope at which a negation signal is active? An intriguing solution to both problems is to develop a simple formula that is mostly right most of the time, a fast and frugal improvement to the approach in our walkthrough that has shown some promise over existing alternatives (e.g., Hii, 2019). However, there is not yet a widely agreed upon formula that experts agree is reliable enough to be trusted across all contexts. Researchers employing these adjustments will still have to perform post hoc validity checks of their results and will still find some (if fewer) cases of mis-scored words in context.

Strategy 5: Utilize a Machine Learning-Based Sentiment Analysis Technique

Recall that all of the sentiment analysis techniques we have discussed here are lexicon-based. They rely on a previously validated dictionary that connects verbal tokens to their associated values. An alternative approach is to use machine learning-based techniques, in which an algorithm attempts to automatically learn how to score new collections of text, based on several (typically large-scale) hand-scored collections it has seen before. This approach has often been shown to be more accurate than lexicon-based approaches alone (e.g., Kotelnikova et al., 2021), but is also beyond the scope of the discussion here. Interested readers are encouraged to consult other related works in this journal (Lanovaz et al., 2020; Turgeon & Lanovaz, 2020) and leading Natural Language Processing conferences (Tang et al., 2019), as well as informative studies that incorporated either lexicon-based sentiment analysis (Yeung et al., 2020; Zhang et al., 2021) or machine-learning based sentiment analysis (Duong et al., 2020; Imtiaz et al., 2022).

Conclusion

Rigorous behavior-analytic research on language, emotion, and other topics of interest in mainstream psychology will require an objective and replicable methodology for quantifying complex verbal behavior (Friman et al., 1998). As described above, a number of these methods

(viz., NLP) are used in other areas of psychology that a) enjoy considerable empirical support and b) are conceptually consistent with Skinnerian and post-Skinnerean accounts of verbal behavior. The methods described in the present discussion are congruent with the foundational dimensions of behavior analysis (Baer et al., 1968), as they rely on direct observation of verbal behavior (and related products; e.g., transcripts), are replicable and well-defined, and have generality to other domains of socially meaningful behavior of interest to behavior analysts. Moreover, this methodology is aligned with calls for additional work in the study of verbal behavior relating to emotion, sentiment, and associated phenomena (Critchfield, Doepke et al., 2017). Fundamentally, this method entails the algorithmic operationalization of verbal operants according to their structural and functional properties. As we have demonstrated, this method can be used in conjunction with other behavior-analytic concepts and models (e.g., the GML) to further examine and quantify response dynamics typically ignored in the realm of behavior analysis. This methodology also has the potential to be leveraged by behavior analysts to understand how behavior-analytic language influences the dissemination of behavioral research and practices to other providers and consumers (Critchfield, Becirevic et al., 2017). A full translation of sentiment analysis techniques into behavior analytic research will require additional studies, not just on the time trade-off that we have illustrated here (see Appendix), but also on the relative accuracy of sentiment analysis compared to the BMC approach - a worthwhile program of research in its own right. However, the failure of behavior analysts to concurrently employ these methods in contemporary research on verbal behavior (e.g., manually coding and quantifying delinquent speech; McDowell & Caron, 2010), may delay future progress in understanding many applied phenomena relating to human communication given the potential of this method for analyzing verbal behavior on a large scale, in real-time, with a highly reliable and replicable procedure.

Data Sharing

Data sharing is not applicable to this article as no datasets were generated or analyzed during the current study.

References

Bailey, J. D., Baker, J. C., Rzeszutek, M. J., & Lanovaz, M. J. (2021). Machine learning for supplementing behavioral assessment. *Perspectives on Behavior Science*, *44*(4), 605–619.

Barnes-Holmes, D., Hayden, E., Barnes-Holmes, Y., & Stewart, I. (2008). The Implicit Relational Assessment Procedure (irap) as a Response-Time and Event-Related-Potentials Methodology for Testing Natural Verbal Relations: A Preliminary Study. *Psychological Record*, *58*(4), 497–515.

Barrie, C., Ho, J. C., Chan, C., Rico, N., König, T., & Davidson, T. (2022). *academictwitteR: Access the Twitter Academic Research Product Track V2 API Endpoint* (0.3.1) [Computer software]. https://CRAN.R-project.org/package=academictwitteR

Becirevic, A., Critchfield, T. S., & Reed, D. D. (2016). On the social acceptability of behavior-analytic terms: Crowdsourced comparisons of lay and technical language. *The Behavior Analyst*, *39*(2), 305-317.

Boyd, R. L., Ashokkumar, A., Seraj, S., & Pennebaker, J. W. (2022). *The Development and Psychometric Properties of LIWC-22*.

Brandt, P. M., & Herzberg, P. Y. (2020). Is a cover letter still needed? Using LIWC to predict application success. *International Journal of Selection and Assessment*, *28*(4), 417–429.

Cero, I., & Witte, T. K. (2020). Assortativity of suicide-related posting on social media. *American Psychologist*, *75*(3), 365–379. https://doi.org/10.1037/amp0000477

Critchfield, T. S., Becirevic, A., & Reed, D. D. (2016). In Skinner's early footsteps: Analyzing verbal behavior in large published corpora. *The Psychological Record*, *66*(4), 639-647.

Critchfield, T. S., & Doepke, K. J. (2018). Emotional overtones of behavior analysis terms in English and five other languages. *Behavior Analysis in Practice*, *11*(2), 97-105.

Critchfield, T. S., Doepke, K. J., Kimberly Epting, L., Becirevic, A., Reed, D. D., Fienup, D. M., ...

& Ecott, C. L. (2017). Normative emotional responses to behavior analysis jargon or how not to use words to win friends and influence people. *Behavior Analysis in Practice*, *10*(2), 97-106.

Cutler, A. D., Carden, S. W., Dorough, H. L., & Holtzman, N. S. (2021). Inferring Grandiose Narcissism From Text: LIWC Versus Machine Learning. *Journal of Language and Social Psychology*, *40*(2), 260–276.

De Choudhury, M., Counts, S., Horvitz, E. J., & Hoff, A. (2014). Characterizing and predicting postpartum depression from shared facebook data. *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing - CSCW '14*, 626–638. https://doi.org/10.1145/2531602.2531675

De Choudhury, M., Kiciman, E., Dredze, M., Coppersmith, G., & Kumar, M. (2016). Discovering Shifts to Suicidal Ideation from Mental Health Content in Social Media. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems . CHI Conference*, *2016*, 2098–2110. https://doi.org/10.1145/2858036.2858207

Dodds, P. S., Harris, K. D., Kloumann, I. M., Bliss, C. A., & Danforth, C. M. (2011). Temporal Patterns of Happiness and Information in a Global Social Network: Hedonometrics and Twitter. *PLoS ONE*, *6*(12), 1–26. https://doi.org/10.1371/journal.pone.0026752

Dragut, E., & Fellbaum, C. (2014, June). The role of adverbs in sentiment analysis. In *Proceedings of Frame Semantics in NLP: A Workshop in Honor of Chuck Fillmore (1929-2014)* (pp. 38-41).

Dragut, E. C., Wang, H., Sistla, P., Yu, C., & Meng, W. (2014). Polarity consistency checking for domain independent sentiment dictionaries. *IEEE Transactions on Knowledge and Data Engineering*, *27*(3), 838-851.

Dubey, A. D. (2020). Twitter sentiment analysis during COVID-19 outbreak. *Available at SSRN 3572023*.

Duong, V., Luo, J., Pham, P., Yang, T., & Wang, Y. (2020). The ivory tower lost: How college students respond differently than the general public to the covid-19 pandemic. *2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 126–130.

Emerson, G., & Declerck, T. (2014, August). SentiMerge: Combining sentiment lexicons in a Bayesian framework. In *Proceedings of Workshop on Lexical and Grammatical Resources for Language Processing* (pp. 30-38).

Friman, P. C., Hayes, S. C., & Wilson, K. G. (1998). Why behavior analysts should study emotion: The example of anxiety. *Journal of Applied Behavior Analysis*, *31*(1), 137–156.

Hawkins, I. I., & Raymond, C. (2022). Measurement of dreams by SCORS and LIWC: Prelude to dreamwork in psychotherapy. *Dreaming*, *32*(1), 52.

Hayes, S. C., Barnes-Holmes, D., & Roche, B. (Eds.). (2001). *Relational Frame Theory: A Post-Skinnerian Account of Human Language and Cognition* (2001 edition). Springer.

Herrnstein, R. J. (1970). On the law of effect. *Journal of the Experimental Analysis of Behavior*, *13*(2), 243–266.

Hii, D. (2019). Using Meaning Specificity to Aid Negation Handling in Sentiment Analysis. *Retrieved November*, *22*, 2020.

Hu, M., & Liu, B. (2004). Mining and summarizing customer reviews. *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 168–177.

Hussey, I., Daly, T., & Barnes-Holmes, D. (2015). Life is Good, But Death Ain't Bad Either: Counter-Intuitive Implicit Biases to Death in a Normative Population. *Psychological Record*, *65*(4), 731–742. https://doi.org/10.1007/s40732-015-0142-3

Imtiaz, A., Khan, D., Lyu, H., & Luo, J. (2022). Taking sides: Public Opinion over the Israel-Palestine Conflict in 2021. *ArXiv Preprint ArXiv:2201.05961*.

Jia, L., Yu, C., & Meng, W. (2009, November). The effect of negation on sentiment analysis and

retrieval effectiveness. In *Proceedings of the 18th ACM conference on Information and knowledge management* (pp. 1827-1830).

Joshi, A., Bhattacharyya, P., & Carman, M. J. (2016). *Automatic Sarcasm Detection: A Survey* (arXiv:1602.03426). arXiv. http://arxiv.org/abs/1602.03426

Jurafsky, D., & Martin, J. (2008). *Speech and Language Processing, 2nd Edition* (2nd edition). Prentice Hall.

Khoo, C. S., & Johnkhan, S. B. (2018). Lexicon-based sentiment analysis: Comparative evaluation of six sentiment lexicons. *Journal of Information Science*, *44*(4), 491–511. https://doi.org/10.1177/0165551517703514

Kiritchenko, S., & Mohammad, S. (2017). The effect of negators, modals, and degree adverbs on sentiment composition. *ArXiv Preprint ArXiv:1712.01794*.

Kotelnikova, A., Paschenko, D., Bochenina, K., & Kotelnikov, E. (2021). Lexicon-based Methods vs. BERT for Text Sentiment Analysis. *ArXiv Preprint ArXiv:2111.10097*.

Lanovaz, M. J., Giannakakos, A. R., & Destras, O. (2020). Machine learning to analyze single-case data: A proof of concept. *Perspectives on Behavior Science*, *43*(1), 21–38.

Lanovaz, M. J., & Hranchuk, K. (2021). Machine learning to analyze single-case graphs: A comparison to visual inspection. *Journal of Applied Behavior Analysis*, *54*(4), 1541–1552.

Liu, B. (2020). *Sentiment Analysis: Mining Opinions, Sentiments, and Emotions* (2nd edition). Cambridge University Press.

Luna, O. (2019). *Matching Analyses as an Evaluative Tool: Characterizing Behavior in Juvenile Residential Settings*.

McDowell, J. J. (2013). On the theoretical and empirical status of the matching law and matching theory. *Psychological Bulletin*, *139*(5), 1000–1028. https://doi.org/10.1037/a0029924

McDowell, J. J., & Caron, M. L. (2010). Matching in an undisturbed natural human environment.

Journal of the Experimental Analysis of Behavior, *93*(3), 415–433.

Meier, T., Boyd, R. L., Pennebaker, J. W., Mehl, M. R., Martin, M., Wolf, M., & Horn, A. B. (2019). “LIWC auf Deutsch”: The development, psychometrics, and introduction of DE-LIWC2015. *PsyArXiv*, *a*.

Mohammad, S., & Turney, P. (2010). Emotions evoked by common words and phrases: Using mechanical turk to create an emotion lexicon. *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, 26–34.

Mohammad, S., & Turney, P. D. (2013a). Crowdsourcing a word–emotion association lexicon. *Computational Intelligence*, *29*(3), 436–465.

Mohammad, S., & Turney, P. D. (2013b). Nrc emotion lexicon. *National Research Council, Canada*, *2*.

Nielsen, F. Å. (2011). *A new ANEW: Evaluation of a word list for sentiment analysis in microblogs* (arXiv:1103.2903). arXiv. https://doi.org/10.48550/arXiv.1103.2903

Normand, M. P., & Donohue, H. E. (2022). Behavior analytic jargon does not seem to influence treatment acceptability ratings. *Journal of Applied Behavior Analysis*, *55*(4), 1294-1305.

Olsson, V., & Lindow, M. (2018). *How does Bipolar and Depressive Diagnoses Reflect in Linguistic Usage on Twitter: A Study using LIWC and Other Tools*.

O’Reilly, A., Roche, B., Ruiz, M., Tyndall, I., & Gavin, A. (2012). The Function Acquisition Speed Test (fast): A Behavior Analytic Implicit Test for Assessing Stimulus Relations. *Psychological Record*, *62*(3), 507–528.

Panicheva, P., & Litvinova, T. (2020). Matching liwc with russian thesauri: An exploratory study. *Conference on Artificial Intelligence and Natural Language*, 181–195.

Pröllochs, N., Feuerriegel, S., & Neumann, D. (2015). Enhancing Sentiment Analysis of Financial News by Detecting Negation Scopes. *2015 48th Hawaii International Conference on System Sciences*, 959–968. https://doi.org/10.1109/HICSS.2015.119

Red Hat. (2020). *What is a REST API?* https://www.redhat.com/en/topics/api/what-is-a-rest-api

Reed, D. D. (2016). *Matching theory applied to MLB team-fan social media interactions: An opportunity for behavior analysis.*

Sarsam, S. M., Al-Samarraie, H., Alzahrani, A. I., Alnumay, W., & Smith, A. P. (2021). A lexicon-based approach to detecting suicide-related messages on Twitter. *Biomedical Signal Processing and Control*, *65*, 102355.

Schneider, A., & Dragut, E. (2015, July). Towards debugging sentiment lexicons. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)* (pp. 1024-1034).

Silge, J., & Robinson, D. (2022). *Text Mining with R: a Tidy Approach* (2022nd-05–03 ed.). https://www.tidytextmining.com/

Simon, C., & Baum, W. M. (2017). Allocation of speech in conversation. *Journal of the Experimental Analysis of Behavior*, *107*(2), 258–278. https://doi.org/10.1002/jeab.249

Skinner, B. F. (1939). Alliteration in Shakespeare's Sonnets: A Study in Liberary Behavior. *The Psychological Record*, *3*, 185.

Skinner, B. F. (1957). *Verbal Behavior*. Copley Publishing Group.

Tang, R., Lu, Y., Liu, L., Mou, L., Vechtomova, O., & Lin, J. (2019). Distilling task-specific knowledge from bert into simple neural networks. *ArXiv Preprint ArXiv:1903.12136*.

Tausczik, Y. R., & Pennebaker, J. W. (2010). The psychological meaning of words: LIWC and computerized text analysis methods. *Journal of Language and Social Psychology*, *29*(1), 24–54.

Taylor, T., & Lanovaz, M. J. (2021). Machine Learning to Support Visual Inspection of Data: A Clinical Application. *Behavior Modification*, 01454455211038208.

Turgeon, S., & Lanovaz, M. J. (2020). Tutorial: Applying machine learning in behavioral research. *Perspectives on Behavior Science*, *43*(4), 697–723.

Twitter. (2022). *Twitter Developer Platform overview*. https://developer.twitter.com/en/docs/platform-overview

Wickham, H., & Grolemund, G. (2017). *R for Data Science: Import, Tidy, Transform, Visualize, and Model Data* (1 edition). O'Reilly Media.

Wickham, H., & RStudio. (2017). *tidyverse: Easily Install and Load the "Tidyverse."* https://CRAN.R-project.org/package=tidyverse

Wong, C. A., Sap, M., Schwartz, A., Town, R., Baker, T., Ungar, L., & Merchant, R. M. (2015). Twitter sentiment predicts Affordable Care Act marketplace enrollment. *Journal of Medical Internet Research*, *17*(2), e3812.

Yeung, N., Lai, J., & Luo, J. (2020). Face off: Polarized public opinions on personal face mask usage during the COVID-19 pandemic. *2020 IEEE International Conference on Big Data (Big Data)*, 4802–4810.

Zhang, H., Gan, W., & Jiang, B. (2014). Machine learning and lexicon based methods for sentiment classification: A survey. *2014 11th Web Information System and Application Conference*, 262–265.

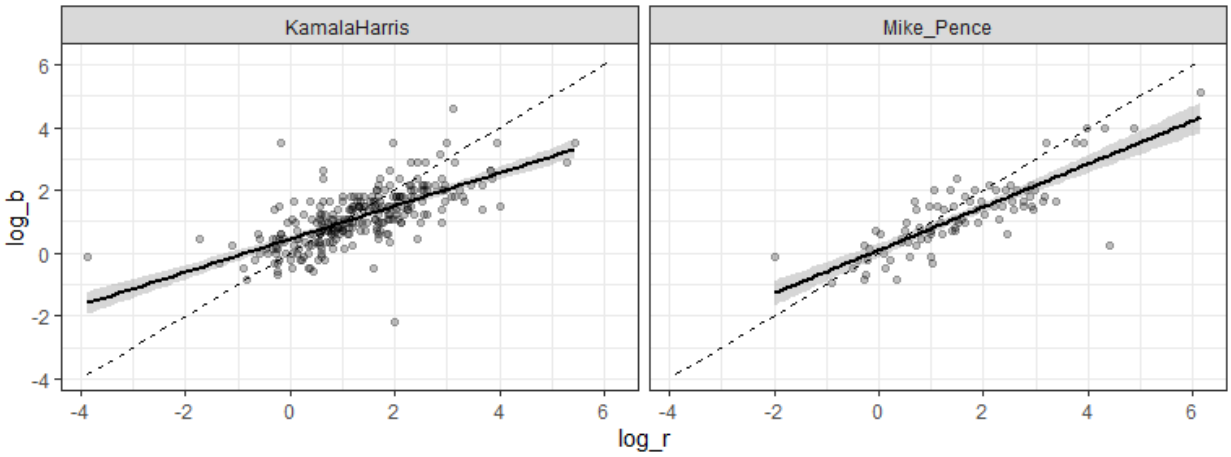
Zhang, X., Wang, Y., Lyu, H., Zhang, Y., Liu, Y., & Luo, J. (2021). The influence of COVID-19 on the well-being of people: Big data methods for capturing the well-being of working adults and protective factors nationwide. *Frontiers in Psychology*, *12*, 2327.

Table 1. *Example Words and Sentiment Values from Three Lexicons*

Word	Lexicon		
	Bing	NRC	AFINN
absolve	-	-	+2
absorbed	-	positive	+1
abundance	positive	anticipation, disgust, joy, negative, positive, trust	-
abundant	positive	-	-
abuse	negative	anger, disgust, fear, negative, sadness	-3

Note. Blank spaces indicate words that are not included in lexicon. Bing is named after one of its creators, Bing Liu. AFINN is also named after one of its creators, Finn Årup Nielsen. NRC = National Research Council in Canada.

Figure 1. *Matching Analyses*



Note. R output for the log2 behavior and reinforcement rates by block for both subjects. Solid lines represent empirically observed regression slopes and gray ribbons represent confidence bounds. Dashed lines represent theoretically perfect matching.

Appendix A

For this worked example, we assume only a basic familiarity with the R programming language and the tidyverse suite of packages within it (Wickham & RStudio, 2017). We have intentionally written the code for maximum readability (sometimes at the cost of brevity), so even readers without this background should still be able to read along. Readers interested in brushing up on R and the tidyverse are encouraged to work through any of the excellent and freely available tutorials available online (e.g., Wickham & Grolemund, 2017).

A basic behaviorally-informed sentiment analysis involves several steps, which we now demonstrate in order.

1. Select a previously validated lexicon or create a new one
2. Acquire raw verbal data (documents)
3. Tokenize your documents and wrangle them into a “tidy” format.
4. Remove stop words / stop tokens
5. Use the lexicon to score each token
6. Compute summary statistics (e.g., proportion of positive words)
7. Analyze with standard behavior analytic methods (e.g., regression, visual analysis)

We will implement these steps to perform an analysis reminiscent of McDowell and Caron’s (2010) work connecting rule-break talk to received praise, in accordance with the GML. Except in this case, we will be examining whether two United States politicians - Vice Presidents Mike Pence and Kamala Harris - post tweets in accordance with the GML.

Step 1: Acquire an Appropriate Lexicon

Technically, steps 1 and 2 can be conducted out of order. We begin with the lexicon in this discussion simply because we needed to begin somewhere. When acquiring a lexicon, a researcher has two options. They can either utilize a pre-validated lexicon from previous research or create a new one. We encourage anyone new to sentiment analysis to use a

pre-validated lexicon, which is both safer and faster. The lexicon you choose can come from a range of sources (Khoo & Johnkhan, 2018). The easiest to use will be those already available in an R package like tidytext (Silge & Robinson, 2022), which includes a helper function to download the lexicons displayed in Table 1. For most of the sentiment analysis a researcher would want to conduct in behavior analysis, these will be sufficient because they include several emotional categories that will get a researcher through their first few studies. By the time a researcher completes their first few studies with these well-known lexicons, they should already have a sense of the kind of things they would want in their next lexicon.

One other lexicon behavioral researchers should be aware of right away, however, is the Linguistic Inquiry and Word Count (LIWC; Boyd et al., 2022; Tausczik & Pennebaker, 2010). This lexicon was created for psychological research and has been evaluated and revised several times. It is especially valuable for its comprehensiveness, including many more word categories than is common in other lexicons (e.g., words related to cognitive processes, social processes, hierarchy). For this reason, LIWC has already been used extensively to study the connection between subjects' linguistic content and a range of psychological topics and in a number of languages (Brandt & Herzberg, 2020; Cutler et al., 2021; Hawkins & Raymond, 2022; Meier et al., 2019; Olsson & Lindow, 2018; Panicheva & Litvinova, 2020). Researchers who find themselves saying "I feel like the basic lexicons aren't enough, I wish I had a lexicon that covered my niche topic" should immediately check whether LIWC covers their particular case.

In this case, we will use the National Research Council Word-Emotion Association Lexicon (NRC) lexicon, which was built up from a range of sources, including the pre-existing WordNet affective lexicon and 8,000 terms from the General Inquirer (Mohammad & Turney, 2010, 2013a). Previous work has used it specifically to study Twitter tweets, including the identification of suicide-related posts, predicting Affordable Care Act enrollment, and to evaluate global pandemic reactions (Dubey, 2020; Sarsam et al., 2021; Wong et al., 2015). This diversity of topics, including one using the NRC to predict overt behavior (e.g., insurance enrollment), all

increase the plausibility this lexicon tracks behaviorally meaningful verbal content. It has the added advantage of being included in the tidytext R package, so we can load it directly like this.

```
library(tidytext)

nrc_lexicon <- get_sentiments(lexicon = 'nrc')

head(nrc_lexicon)
```

```
## # A tibble: 6 x 2
##   word      sentiment
##   <chr>     <chr>
## 1 abacus    trust
## 2 abandon   fear
## 3 abandon   negative
## 4 abandon   sadness
## 5 abandoned anger
## 6 abandoned fear
```

With the NRC lexicon loaded into memory, we can narrow down the kind of sentiment we want to study in this analysis. Here, we retain only words that are related to the dimension of trust / mistrust. We expect this dimension is especially relevant to the occupational success of our two subjects, so it is likely to be a function of some salient reinforcer - like the number of “likes” from Twitter followers. Below, we also provide a random sample of the remaining trust-related words.

```
library(tidyverse)

nrc_trust <- nrc_lexicon %>%
  filter(sentiment == 'trust')

nrc_trust %>%
  sample_n(5)
```

```
## # A tibble: 5 x 2
##   word        sentiment
##   <chr>       <chr>
## 1 protected   trust
## 2 deceiving   trust
## 3 vouch       trust
## 4 ballot      trust
## 5 considerate trust
```

Step 2: Acquire Raw Verbal Data

Most researchers will already be aware of verbal data sources relevant to their research (e.g., intervention session transcripts), so we will avoid repeating some of the most common

sources here. Instead, we point out that there are likely a few data sources that researchers have not previously considered. For example, some video conference platforms (e.g., Zoom) have built-in support for automatic transcription of recorded meetings. And readers will be pleased to learn that these transcriptions are both quite accurate and arrive in a standardized format. In a yet unpublished study, our own research group has already taken advantage of these resources, finding that the latency, duration, and content of speech is associated with intervention satisfaction, recall, and self-reported adoption at one-month followup (manuscript in preparation).

Another example is Project Gutenberg, which provides digital versions of public domain literature. Although this is outside the scope of most modern behavior studies, we mention it to interested readers who might want to follow in Skinner's early footsteps, which actually began with an analysis of alliteration in the works of William Shakespear (Skinner, 1939).

The last approach - and the one we use for our worked example - is to use a REST API.[2] Usually shortened to just "API", this is a system for communicating with a web server via code, rather than a point-and-click interface. This process requires some initial effort, but is often simpler than it sounds and is a quick way to access a substantial amount of data. One of the most well known APIs in research is the Twitter API, which allows people outside of Twitter to access a substantial amount of granular data on the activity of Twitter's users. To give readers a sense of the scope, the first author was able to gather 64 million tweets from 17 million different for a recent study - all for free (Cero & Witte, 2020).

Although a comprehensive introduction to APIs is beyond the scope of the current discussion, Twitter's own tutorial is a great introduction and will remain up-to-date whenever they implement changes (Twitter, 2022). In practice, the process involves filling out a brief application to Twitter, who will then provide a set of tokens that function like a username and

[2] This intimidating acronym stands for Representational State Transfer Application Programming interface (Red Hat, 2020), but that is not especially informative to most readers.

password. Researchers can then pass these tokens and a search query to an R package that knows how to handle the Twitter API (e.g., academictwitteR; Barrie et al., 2022), doing most of the work under the hood.

For example, to save the roughly 30,000 posts Pence and Harris have produced from 2016 through 2021, a researcher simply provides their bearer token from Twitter, a formatted search query for tweets from Pence's and Harris' accounts, the dates to search through, and a data path (folder) in which to save the results.

```
library(academictwitteR)

btoken = 'YOUR TOKEN HERE'
data_folder <- 'data/presidential_tweets'

vp_tweets <- academictwitteR::get_all_tweets(
  query = 'from:Mike_Pence OR from:KamalaHarris',
  start_tweets = '2016-01-01T00:00:00.000Z',
  end_tweets = '2022-01-01T00:00:00Z',
  n = 50000,
  data_path = data_folder,
  bind_tweets = F,
  bearer_token = btoken)
```

The `get_all_tweets()` function unfortunately saves tweet and user data in separate places, so we'll need to load and merge them ourselves. For the loading, we can use the `bind_tweets()` function from the `academictwitteR` package to bring the tweets and user data into memory. Along the way, we extract (`unnest(public_metrics)`) some information about each tweet, including the `like_count` - our putative reinforcer in this mini matching study. We'll also filter out retweets (which always start with an "RT"), retaining only the tweets generated by Pence and Harris themselves. By coincidence, this leaves exactly 18,000 tweets in total.

We can then use the `left_join()` function of the `tidyverse` package to add user information to each tweet.

```
tweets_df <- bind_tweets(data_folder) %>%
  unnest(public_metrics) %>%
  filter(substr(text, 1, 2) != 'RT') %>%
  select(author_id, tweet_id = id, like_count, text)
```

```
##
================================================================================
==

user_df <- bind_tweets(data_folder, user = T) %>%
  select(author_id = id, username) %>%
  unique()

##
================================================================================
==

full_df <- left_join(tweets_df, user_df)

## # A tibble: 6 x 4
##   username     tweet_id            like_count text
##   <chr>        <chr>                    <int> <chr>
## 1 KamalaHarris 1025194038490787842       2294 "These repeated attacks are
dange~
## 2 KamalaHarris 1025173908373286913       1194 "One of the most critical
issues ~
## 3 KamalaHarris 1025162835284951040       7170 "Russia attacked our country
duri~
## 4 KamalaHarris 1025134153145221121       1828 "Today we remember author and
civ~
## 5 KamalaHarris 1025128359880220672       1896 "California is actively
working t~
## 6 KamalaHarris 1025102206767427584        513 "As wages have largely
remained s~
```

Step 3: Tokenize Your Documents and Wrangle Them into a "Tidy" Format

In its current form, our `full_df` dataframe stores each tweet as a line of text. Although it is easy to read, this makes it hard for our code to access each individual word and compare it to the entries in our NRC lexicon. To get around this, we need to tokenize all of our tweets, so that each row of our dataframe will represent a single word. This is called the *tidy* format in R. Fortunately, the `tidytext` package makes this process easy, providing us with the `unnest_tokens()` function that handles everything automatically. We simply tell it we want a new column named `word`, which is made up of the individual words from the old `text` column. Careful readers will thus notice the first several entries in the `word` column of the `tokenized_df` now represent the first several words of the first text in the `text` column of the `full_df`.

```
tokenized_df <- full_df %>%
  unnest_tokens(word, text)
```

```
## # A tibble: 5 x 5
##   author_id tweet_id            like_count username     word
##   <chr>     <chr>                    <int> <chr>        <chr>
## 1 30354991  1025194038490787842       2294 KamalaHarris these
## 2 30354991  1025194038490787842       2294 KamalaHarris repeated
## 3 30354991  1025194038490787842       2294 KamalaHarris attacks
## 4 30354991  1025194038490787842       2294 KamalaHarris are
## 5 30354991  1025194038490787842       2294 KamalaHarris dangerous
```

Step 4: Remove Stopwords from the Dataset

Stop words or stop tokens (in the case of multiword n-grams) are those that occur so often that they are uninformative to the meaning of a text (e.g., “of”, “and”, “the”). Fortunately, just by loading the `tidytext` package, we have already loaded a pre-compiled list of stopwords called `stop_words` in the background. Thus, the quickest way to get these stopwords out of our `tokenized_df` dataframe is simply to `anti_join()` them. In an anti-join or anti-merge, only the records from the first dataset (`tokenized_df`) that DO NOT match anything in the second dataset (`stop_words`) are retained.

While we are removing unhelpful tokens, we’ll also filter out “t.co” and “https”. Visual inspection of our `tokenized_df` revealed these are both fragments of web links Pence and Harris posted in some of their tweets, which were accidentally included during the tokenization process (`unnest_tokens()` thought they were words worth retaining). Because our lexicon does not cover them, we can explicitly filter them out here too.

```
tokenized_df <- tokenized_df %>%
  anti_join(stop_words) %>%
  filter(! word %in% c('t.co', 'https'))
```

```
## # A tibble: 5 x 5
##   author_id tweet_id            like_count username     word
##   <chr>     <chr>                    <int> <chr>        <chr>
## 1 30354991  1025194038490787842       2294 KamalaHarris repeated
## 2 30354991  1025194038490787842       2294 KamalaHarris attacks
## 3 30354991  1025194038490787842       2294 KamalaHarris dangerous
## 4 30354991  1025194038490787842       2294 KamalaHarris democracy
## 5 30354991  1025194038490787842       2294 KamalaHarris congress
```

Step 5: Use the Lexicon to Score Each Token

We expect this likely sounds as though it will be the most intensive part of sentiment analysis. After all, we estimated for McDowell and Caron's group to hand-score a much smaller sample of text likely took over 140 person hours. Scoring all the words from 18,000 tweets must be quite laborious, right? In fact, all of our words are effectively scored with just two lines of code, which join the words from our observed dataset to the values in our NRC trust lexicon.

```
sentiment_df <- tokenized_df %>%
  left_join(nrc_trust)
```

```
## # A tibble: 5 x 4
##   author_id tweet_id            word      sentiment
##   <chr>     <chr>               <chr>     <chr>
## 1 30354991  1025194038490787842 repeated  <NA>
## 2 30354991  1025194038490787842 attacks   <NA>
## 3 30354991  1025194038490787842 dangerous <NA>
## 4 30354991  1025194038490787842 democracy <NA>
## 5 30354991  1025194038490787842 congress  trust
```

A minor snag is that our `nrc_trust` lexicon only includes words that are trust-related. It produces missing values for everything on which the lexicon is silent (i.e., non-trust words). To simplify our upcoming analysis, we'll compute a new True/False column called `trust_word`, which will indicate whether a given word in our dataset is a trust word, based on the values in the adjacent `sentiment` column.

```
sentiment_df <- sentiment_df %>%
  mutate(trust_word = ifelse(is.na(sentiment), F, T))
```

```
## # A tibble: 5 x 5
##   author_id tweet_id            word      sentiment trust_word
##   <chr>     <chr>               <chr>     <chr>     <lgl>
## 1 30354991  1025194038490787842 repeated  <NA>      FALSE
## 2 30354991  1025194038490787842 attacks   <NA>      FALSE
## 3 30354991  1025194038490787842 dangerous <NA>      FALSE
## 4 30354991  1025194038490787842 democracy <NA>      FALSE
## 5 30354991  1025194038490787842 congress  trust     TRUE
```

As a quick sanity check, we'll now peak at a random sample of trust and non-trust words from both subjects.

```
sentiment_df %>%
  group_by(username, trust_word) %>%
```

```
  sample_n(size = 2) %>%
  select(word)
```

```
## # A tibble: 8 x 3
## # Groups:   username, trust_word [4]
##   username     trust_word word
##   <chr>        <lgl>      <chr>
## 1 KamalaHarris FALSE      local
## 2 KamalaHarris FALSE      video
## 3 KamalaHarris TRUE       system
## 4 KamalaHarris TRUE       safe
## 5 Mike_Pence   FALSE      persecution
## 6 Mike_Pence   FALSE      fair
## 7 Mike_Pence   TRUE       system
## 8 Mike_Pence   TRUE       proud
```

Here, we get a quick sense of the kinds of trust- and nontrust- related words each subject might be using. These randomly selected words are overall somewhat banal, but they are consistent with what we would expect. Words like “system” imply something that needs to be relied on, so they exist somewhere along a dimension of trustworthiness. Words like “fair” are morally-salient, but do not imply something related to reliance and thus not scored as trust-related. The same is true of words like persecution, which is unfair to be sure, but does not indicate a dimension of trust.

Step 6: Compute Summary Statistics

For our upcoming matching analysis, we’ll want to know whether each subject produces tweets with trust-related words in proportion to the likes those tweets received. To get this far, we needed to break up (“tokenize”) whole tweets into individual words, so that we could score those words with a lexicon. Now that they have been scored in the `trust_word` column, we need to start going in reverse. We need to recombine words into tweets and summarize each tweet by whether any of its words is a trust word.

```
summary_df <- sentiment_df %>%
  group_by(username, tweet_id, like_count) %>%
  summarize(is_trust_tweet = any(trust_word))
```

```
## # A tibble: 4 x 4
## # Groups:   username, tweet_id [4]
##   username     tweet_id            like_count is_trust_tweet
```

```
##   <chr>        <chr>                    <int> <lgl>
## 1 KamalaHarris 1000061607630856193        590 FALSE
## 2 KamalaHarris 1000080062887157760       2680 TRUE
## 3 KamalaHarris 1000133158799593473        768 FALSE
## 4 KamalaHarris 1000145751706660864        995 TRUE
```

Once individual tweets have been scored in the `is_trust_tweet` column, we arrange tweets by their ID numbers (which are strictly in order of date produced) and assign them to blocks of 50 people. This final line, `block = floor(row_number() - 1) / 50`, is just a shorthand way of saying “take the row number of each tweet, divide by 50, and round down to the nearest integer, and treat that as its block number.”

```
blocked_df <- summary_df %>%
  group_by(username) %>%
  arrange(tweet_id) %>%
  mutate(block = floor((row_number() - 1) / 50))
```

With blocks assigned, we simply compute the familiar matching statistics. One trick to note is that R will treat `TRUE` and `FALSE` values as `1` and `0` when they are forced into mathematical computations. Thus, `sum(like_count*is_trust_tweet)` can be read “the sum of likes produced when `is_trust_tweet` is true.” Proactively, we also `filter()` to retain only cases where the `log_b` and `log_r` are still finite, which in this case is all of them because there were no blocks with 0 trust/non-trust tweets or 0 likes for either of those cases.

```
matching_df <- blocked_df %>%
  group_by(username, block) %>%
  summarize(
    b1 = sum(is_trust_tweet),
    b2 = sum(!is_trust_tweet),
    r1 = sum(like_count*is_trust_tweet),
    r2 = sum(like_count*(!is_trust_tweet)),
    log_b = log2(b1/b2),
    log_r = log2(r1/r2)) %>%
  filter(is.finite(log_b) & is.finite(log_r))
```

```
## # A tibble: 5 x 8
## # Groups:   username [1]
##   username     block    b1    b2     r1    r2 log_b log_r
##   <chr>        <dbl> <int> <int>  <int> <int> <dbl> <dbl>
## 1 KamalaHarris     0    30    20 121162 32965 0.585  1.88
## 2 KamalaHarris     1    43     7 130317  9112 2.62   3.84
## 3 KamalaHarris     2    41     9 117068 36830 2.19   1.67
```

```
## 4 KamalaHarris     3    38    12  93230 30820 1.66   1.60
## 5 KamalaHarris     4    39    11 362427 89177 1.83   2.02
```

Step 7: Analyze with Standard Behavior-Analytic Methods (e.g., Visual Analysis, Regression)

At this point, all that is left to do is perform a matching analysis. Because we have two subjects who will need separate regressions, we use the group-nest-map-tidy-unnest approach. It is probably overkill for only two regressions, but in the common case that a matching analysis includes a half-dozen or more subjects to regress, this strategy is both faster and safer than copy-pasting code.

```
model_df <- matching_df %>%
  group_by(username) %>%
  nest() %>%
  mutate(
    fitted_model = map(data, ~ lm(log_r ~ log_b, data = .x)),
    model_coefs = map(fitted_model, broom::tidy),
    model_stats = map(fitted_model, broom::glance))
```

Unnesting the regression coefficients reveals some interesting results. Both subjects are highly sensitive to the likes associated with trust-related words.

```
model_df %>%
  unnest(model_coefs) %>%
  mutate(term = ifelse(term == '(Intercept)', 'bias', 'sensitivity')) %>%
  select(username, term, estimate, std.error, p.value) %>%
  mutate(across(where(is.numeric), .fns = ~ round(.x, 3)))
```

```
## # A tibble: 4 x 5
## # Groups:   username [2]
##   username     term        estimate std.error p.value
##   <chr>        <chr>          <dbl>     <dbl>   <dbl>
## 1 KamalaHarris bias           0.278     0.083   0.001
## 2 KamalaHarris sensitivity    0.892     0.058   0
## 3 Mike_Pence   bias           0.419     0.123   0.001
## 4 Mike_Pence   sensitivity    0.989     0.074   0
```

What is even more interesting are these substantial bias terms, which suggest that even if trust-related tweets produced likes in equal proportion to nontrust-related tweets, both subjects would still produce trust-related tweets in substantial excess. Specifically, even if the likes received for each tweet type were perfectly balanced, Harris would be expected to produce

2^0.278 = 1.21x more trust-related tweets than nontrust related ones - and Pence would produce 1.34x more.

Examining the efficacy of the matching model to explain such behavior, note that the R-squared values for both subjects are significant, but much higher for Pence. Combined with a sensitivity very near 1.0 for this subject, such a finding suggests this learning model is a compelling (if as yet, non-experimental) account of his verbal behavior over many years.

```
model_df %>%
  unnest(model_stats) %>%
  select(username, r.squared, p.value) %>%
  mutate(across(where(is.numeric), .fns = ~ round(.x, 3)))
```

```
## # A tibble: 2 x 3
## # Groups:   username [2]
##   username     r.squared p.value
##   <chr>            <dbl>   <dbl>
## 1 KamalaHarris     0.47        0
## 2 Mike_Pence       0.679       0
```

We can see this by visually examining the log behavior and log reinforcement rates for each subject for each block, observing that Pence's blocks conform much more closely to the theoretically perfect matching (the dashed line).

```
ggplot(matching_df, aes(log_r, log_b)) +
  facet_grid(. ~ username) +
  geom_point(alpha = .25) +
  geom_smooth(method = 'lm', color = 'black') +
  stat_function(fun = function(x) 1*x, linetype = 2) +
  theme_bw()
```