Human computation scaling for measuring meaningful latent traits in political texts[*]

Jacob M. Montgomery
Washington University in St. Louis

David Carlson
Washington University in St. Louis

July 21, 2016

ABSTRACT

Political scientists are increasingly interested in measuring latent political concepts embedded in written or spoken records. After all, most important political behaviors and outcomes are encoded in language. However, current approaches of turning natural language into meaningful measures are sometimes unsatisfying, relying on either costly and unreliable human coding or automated methods for document classification that miss subtleties of language easily identified by human readers. In this paper, we develop and validate an innovative "human computation" method for encoding political texts that preserves much of the reliability of automated methods while leveraging the superior ability of humans to read and understand natural language. We validate the method with online movie reviews, open-ended survey responses, advertisements for U.S. Senate candidates, and State Department reports on human rights. The framework we present is quite general, and we provide software to help researchers interact easily with online workforces to extract meaningful data from texts.

[*] We thank John Freeman and Brandon Stewart for providing comments on a previous version of this paper. We are indebted to Ryden Butler, Dominic Jarkey, Jon Rogowski, Erin Rossiter, and Michelle Torres for their assistance with this project. We particularly wish to thank Matt Dickenson for his programming assistance. We also appreciate the assistance in the R package development from David Flasterstein, Joseph Ludmir, and Taishi Muraoka. We are grateful for the financial support provided by Steve Smith and the Weidenbaum Center on the Economy, Government, and Public Policy. Finally, we wish to thank the partner-workers at Amazon's Mechanical Turk who make this research possible.

1 INTRODUCTION

Given the centrality of words to the political process, it is unsurprising that political scientists have always been deeply interested in studying and characterizing the content of spoken and written language. These records are, after all, often the most direct evidence we have about the true nature of political debates, the intentions of political actors, and the policy outcomes reached by political institutions. Indeed, political science in earlier eras rested heavily on the analysis of language originating from interviews (Key 1949; Kingdon 1973), participant observation (Fenno 1978) and more. In fact, in several areas of inquiry, a heavy focus on language persists. Projects such as the Comparative Manifesto Project (Budge 2001) and the Policy Agendas Projects (Baumgartner and Jones 1993) represent massive endeavors to characterize and explain the nature of important political texts and speeches.

Yet, considering the broad contours of standard practices in social science research over the past six decades, the systematic study of natural language has declined precipitously, largely in conjunction with the rise of statistically oriented forms of inquiry. Today, political scientists seeking to test theories largely ignore what is *said or written* and instead focus on more easily quantifiable *behaviors* (e.g., roll-call votes). While political science has advanced significantly relying on behavioral indicators, they do not represent the wholeness of our political realities. On the contrary, few students of politics would deny that the discipline's near-exclusive reliance on easily quantifiable metrics impoverishes its engagement with the realities of politics in practice. It is as if Lincoln's ascendancy to the presidency can be captured by the mere fact of his inauguration rather than his appeal to the, "better angels of our nature," the outcome of *Brown v. The Topeka Board of Education* is fully encapsulated as a vote total rather than an announcement that, "Separate educational facilities are inherently unequal," or that Senator Obama's loss in the 2008 New Hampshire primary can be summarized in the outcome rather than his proclamation that, "Yes we can."

While the subsidiarity of language to other measures in recent decades likely has many explanations, certainly one contributing factor is the difficulty of turning naturally generated human language into quantitative measures that retain their substantive and theoretical value. Broadly speaking, researchers analyzing the content of written or spoken language have either used expert coders (e.g., trained research assistants) or one of several text-analysis algorithms derived from computer science (e.g., topic models). However, both expert coding and machine learning algorithms run afoul of the multi-valued and interpretative nature of human communication. The meaning of words is subjective and interpretable only within a specific context, particularly in the political realm where the very definitions of terms are points of conflict and debate (e.g., "marriage"). So, for instance, the claim that humans have a "right to life" indicates freedom from state-sponsored violence in the context of Article 3 of the *Universal Declaration of Human Rights*, but is a justification for the violent annexation of territory in the context of 1920s German politics.[1] The result of the inherent contextual and subjective qualities of natural language means that quantifying the underlying meaning of any text is to some degree interpretive.

Why is the interpretive nature of natural language an obstacle to generating meaningful quantitative measures? It is extremely difficult to turn language into measures that are simultaneously reliable and valid indicators of important political concepts. Human coders are able to easily read a sentence and assign political meaning, but subtle differences in the underlying interpretations of a text lead to low levels of inter-coder reliability. Rigid coding rules focused on less interpretative aspects of texts are sometimes able to improve reliability, but at the cost of focusing on features that are intentionally less subjective and, as a consequence, of less direct interest.

Recently, many scholars have turned to the analysis of political texts with automated techniques coming from computer science. Despite their promise, however, the outputs of these models may not reflect the underlying concepts of interest to researchers and can be difficult to interpret. As Grimmer and Stewart (2013, p. 271) note, "Automated text analysis methods can substantially reduce the costs and time of analyzing massive collections of political texts. When applied to any one problem, however, the output of the models may be misleading or simply wrong." Moreover, relatively little work to date has focused on measuring continuous latent traits. While there are several important exceptions (e.g., Laver, Benoit, and Garry 2003; Slapin and Proksch 2008), many

[1] "The first right in the world is the right to life, provided one has the strength for it" (Hitler 2013, p. 18).

prominent studies have instead focused on classifying documents into unordered categories (e.g., Quinn et al. 2010; Grimmer 2010).

In this paper, we draw on insights from the field of human computation to propose a method that combines the advantages of both the traditional content-analysis and automated approaches for turning language into data. We develop and validate a general-purpose framework for encoding natural language by dividing the larger undertaking into thousands of simple micro-tasks (viz., binary pairwise comparisons) that can be easily completed by a trained but non-expert online workforce. By sending thousands of these simple comparisons to online workers, we circumvent issues of unreliable human coding while capitalizing on the superior abilities of humans to understand context and sentiment in natural language. We then statistically post-process the resulting data to construct valid and reliable estimates of latent traits within documents. Importantly, the output of our method is not a categorization, but a meaningful measure scored on a continuous scale.

After providing the details of our method, we evaluate it using texts from online movie reviews, open-ended survey responses, congressional advertisements, and State Department reports. In each case, we rely on alternative measures of researcher-specified traits in each document to show that the resulting measures are not only highly reliable, but valid measures of underlying latent traits of interest. The framework we present is quite general, and we provide software and practical guidelines to help researchers smoothly interact with online workforces.

2 THE CHALLENGE OF ENCODING NATURAL LANGUAGE

Until very recently, the most common approach to coding language was to conduct some form of content analysis (e.g., Krippendorff 2013). This involves, first, dividing some text into units (e.g., paragraphs) to be analyzed, and, second, placing each unit into a category based on a coding rubric. While intuitive and conceptually straightforward, turning texts into meaningful quantitative data is plagued from two well-known weaknesses. First, it can be prohibitively expensive even for modestly sized collections of documents. Coding documents is often a tedious task, but one which demands a certain level of training and expertise to perform correctly. Finding, training, and compensating research assistants can be time-consuming and costly.

For instance, the most comprehensive effort to date that has assembled information on campaign strategy is provided by Druckman, Kifer, and Parkin (2009), who collected information from the websites of a random sample of major-party candidates for the U.S. House of Representatives in the ten days preceding the 2002, 2004, and 2006 general elections. But this effort itself reveals the arduous nature of this task using traditional methods. The authors are able to code *only* major-party Senate candidates and a random sample of 20% of House candidates.

Second, and more problematic, past experience shows that even highly trained and competent coders provide unreliable estimates when asked to code even modestly subjective topics. Mikhaylov, Laver, and Benoit (2012), for instance, investigate the inter-coder reliability of categorizations produced by expert coding of party manifestos. They find that the existing coding process, "is prone to unacceptably high levels of unreliability " (p. 90). They conclude that, "[T]he propensity. . . for misclassification by human coders, even trained and experienced coders, suggests a need for a much simplified coding scheme" (p. 90).

This trade-off between the subtlety of the coding scheme and the reliability of the measure is well known. We might, for instance, wish to code the 'tone' of statements on candidate websites on a 100-point scale ranging from strongly negative to strongly positive. However, implementing such a scheme is simply beyond the capacity of human coders to execute reliably and is flatly impossible when multiple coders are involved (Krosnick 1999; Oishi et al. 2005). Thus, Druckman, Kifer, and Parkin (2009) coded candidate websites on characteristics easily identifiable to individual research assistants such as whether the website provides a candidate biography, mentions her family, mentions her opponent, or includes polling information. In the end, despite requiring many hours of human labor to build, the dataset contains nothing about the tone, ideological content, or even the stated policy positions of candidates.

Given the expense and difficulty of manually coding language, it is unsurprising that political scientists have embraced advances in computer science that allow for automated coding. Recent work drawing on this family of methods has been used to study congressional speech (Monroe, Colaresi, and Quinn 2008), categorize open-ended survey responses (Roberts et al. 2014), understand the representational style of elected officials (Grimmer 2013), and more. Despite the obvious advantages of automated methods relative to relying on trained research assistants in terms of reliability and cost, it is important to understand their fundamental limitations. First, as most practitioners of text analysis acknowledge, the outputs of these models are dramatic simplifications of the underlying language. As Grimmer and Stewart (2013) note, "The complexity of language implies that all methods necessarily fail to provide an accurate account of the data-generating process used to produce texts" (p. 270). Virtually all text models applied in a political context are variants of "bag of words" methods that strip language of not only context but even word ordering, punctuation, and tense. This does not mean that text models are without value – far from it – but rather that they are not appropriate for capturing all aspects of language that may be of interest to researchers. More centrally, this reductive approach means that text models struggle to differentiate between statements that are trivial for even untrained human coders to distinguish.

A second limitation of many of the dominant text models in the literature is that they are focused on partitioning documents into distinct clusters or grouping (e.g., Hopkins and King 2010). While classification is certainly valuable for answering some questions, researchers are often more interested in scaling documents to extract measures of researcher-specified underlying traits.[2]

Scaling continuous latent traits in text is certainly not unknown in political science. The most prominent example is the `WordScore` model proposed by Laver, Benoit, and Garry (2003) for placing party manifestos on an ideological scale. Other notable examples include the `WordFish` model (Slapin and Proksch 2008) and dictionary-based methods (e.g., Owens and Wedeking 2012).[3] However, we argue that the general applicability of these methods across domains is limited. To begin with, there are concerns that – at least in some cases – these methods provide low-quality estimates. For example, Lowe and Benoit (2013) benchmark the `WordFish` method for scaling ideology in texts against expert human coders and found that in many cases the statistical methods

[2] While it is possible to partially equate unsupervised topic models with scaling models (e.g., Pang and Lee 2005), the process is likely to be frustrating and largely unproductive (Grimmer and Stewart 2013, p. 281).

[3] Other examples might include Lowe et al. (2011) or Jamal et al. (2015) depending on how one interprets the output. Another branch of recent research seeks to combine texts with ancillary information to provide unsupervised sentiment-scaling of speeches. Several recent papers, for instance, combine text with roll call votes to measure ideology (e.g., Lauderdale and Herzog 2014; Kim, Londregan, and Ratkovic 2014).

were wildly inaccurate. Grimmer and Stewart (2013, p. 293) further show that the underlying meaning of `WordFish` scores changes radically depending on the content of the document set. Likewise, Budge and Pennings (2007) use `WordScores` to code speeches in the Irish Parliament to measure party support for the budget. While the automated methods placed Sinn Féin in the middle of the spectrum, all human coders were able to easily place them at the political extreme.

Further, a fundamental limitation of both supervised learning and dictionary-based approaches is that they assume the existence of a well-validated document set, something rarely available in political science. Supervised methods “learn” how specific word frequencies are associated with an underlying trait by estimating the relationship between words and measured traits in training datasets. Laver, Benoit, and Garry (2003), for instance, train their model using a set of expert-coded party manifestos. In the end, the validity of the resulting measure rests entirely on the quality of the training set. This brings us full circle to the difficulty of using human experts to reliably code complex documents or build dictionaries. Likewise, dictionary-based methods assume that the meaning (or valence) of a specific word is consistent with its assigned valence in the dictionary, which is itself developed within a specific research domain. However, “when dictionaries are created in one substantive area and then applied to another, serious errors can occur” (Grimmer and Stewart 2013, p. 274).

The method we propose relies neither on statistical methods nor human coding alone. The intuition is that we can combine the superior ability of humans to read and understand the meaning of natural language with the superior ability of computers to aggregate data into reliable measures of latent traits. As we demonstrate, this combination allows us to produce valid measures of latent concepts embedded in texts that better reflect the complexity of human communication. Further, the traits of interest can be specified in advance by the researcher, estimates exist on a continuous scale, and the measures are highly reliable.

Before presenting our method, it is important to note that the idea of using online workforces to code text is not unknown in political science research. Henderson (2015), for example, used online workforces to guess the ideological origins of political ads. However, by far the most similar approach to our own is presented by Benoit et al. (Forthcoming), who develop an approach to coding party manifestos using online workforces. The `SentimentIt` system we present below differs in that our aim is not to provide a method for encoding a specific corpus of texts (e.g., party manifestos), but rather to provide a general framework for creating reliable and valid measures of latent traits for a wide array of document sets. We show that our framework can be applied in areas ranging from online posts, to surveys responses, to political advertisements, to reports from bureaucratic agencies. Further, the `SentimentIt` software we provide offers a suite of tools for researchers to smoothly interact with online workforces, manage workers, and analyze data, all within the increasingly common `R` computing environment.

3 THE `SENITMENTIT` SYSTEM FOR HUMAN COMPUTATION TASKS

Human computation (HC) is the study of algorithms and procedures that integrate the innate abilities of humans with the developing capabilities of computer algorithms to together solve problems that neither humans nor computers can handle alone (Quinn and Bederson 2011). HC as a field was pioneered by Luis von Ahn, who, in summarizing his research, states:

> Although computers have advanced dramatically over the last 50 years, they still do not possess basic conceptual intelligence or perceptual capabilities that most humans take for granted. By leveraging human abilities ... I solve large-scale computational problems and collect data to teach computers basic human talents. To this end, I treat human brains as processors in a distributed system, each performing a small part of a massive computation. (Von Ahn 2009)

While a wide variety of methods fit under this umbrella,[4] HC methods have three basic components (Quinn and Bederson 2011). First, the researcher must create and organize a corpus of specific texts or images that need to be analyzed. Second, HC makes use of large online workforces to perform small evaluative tasks. For example, Von Ahn et al. (2008) used the `ReCAPTCHA` security protocol to have humans recognize words in non-digitized texts that computers could not confidently identify. Finally, human evaluations are aggregated in some fashion to create an output

[4] See Quinn and Bederson (2011) for further examples and a detailed discussion distinguishing HC from related concepts such as crowdsourcing and social computing.

useful for the end-user. Typically, this is done using statistical post-processing, redundancy, or other methods that ensure that some degree of human error is removed.

3.1. *Design principles*

Following the principles laid out in Quinn and Bederson (2011), we designed our system, which we label `SentimentIt`, based on the following criteria. First, `SentimentIt` leverages the *human ability* to understand language and socially constructed political concepts. Importantly, while all of our tasks are to some degree subjective, we focus on human ability to discern pre-defined characteristics embedded within text (e.g., positivity) rather than explicitly subjective characteristics (e.g., persuasiveness). That is, we endeavor to focus on characteristics that can be defined clearly and are therefore less subject to coder-specific biases.

Second, we designed the *task structure* to be cognitively appropriate for non-experts. Specifically, we ask workers to conduct pairwise comparisons of texts, simply indicating which text is more extreme along a single dimension of interest (e.g., "Which text is more positive?"). A significant body of research indicates that pairwise comparisons can reduce the cognitive burden for respondents, improve the reliability of responses, and eliminate problems such as differential item functioning and reference group effects that plague alternative question formats (Brady 1985; King et al. 2004; Oishi et al. 2005). Note that this distinguishes `SentimentIt` from previous approaches for using online work forces to analyze text, which rely exclusively on Likert-format questions where texts are presented sequentially in isolation (e.g. Benoit et al. Forthcoming). In addition to the results below, we provide evidence that workers are able to reliably complete pairwise comparison tasks measuring an unidimensional concept in Appendix SI-5.

A third design principal is the *motivation* of workers. In this case, we relied on paid online workers recruited through Amazon's Mechanical Turk (AMT). In our examples below, we pay between \$0.04 and \$0.10 for each pairwise comparison.[5] While it is now common for researchers to use AMT workers as research subjects (Berinsky, Huber, and Lenz 2012), the majority of jobs

[5] One limitation of AMT workers is that they are increasingly drawn exclusively from U.S. citizens, limiting its applicability to only English-language texts. However, the `SentimentIt` system is designed to be able to integrate with multiple online workforces, and we plant to extend it to allow researchers to post micro-tasks internationally.

posted at AMT are actually completing HITs (human intelligence tasks). AMT workers are experienced at completing micro-tasks and, as we demonstrate below and in our online appendices, capable of quickly providing high-quality data at a very low cost.

Our final design principal is *quality assurance* via training, redundancy, and statistical monitoring. To begin, in order to qualify for our micro-tasks workers must complete an online training module that explains the task, provides detailed decision rules, and provides example tasks with detailed discussion of difficult cases. As part of this training, workers must correctly complete a preset number of tasks before they are "certified" to participate.[6]

We also rely on redundancy. Each document is included in multiple pairwise comparisons. We found that including each document in 20 pairwise comparisons provides very high quality estimates, although deviations from this suggestion may be appropriate on a case by case basis. The concept of redundancy builds on extensive research illustrating that aggregated judgments by non-experts are often comparable to those provided by subject experts (e.g., Benoit et al. Forthcoming; Snow et al. 2008; Sheng, Provost, and Ipeirotis 2008).

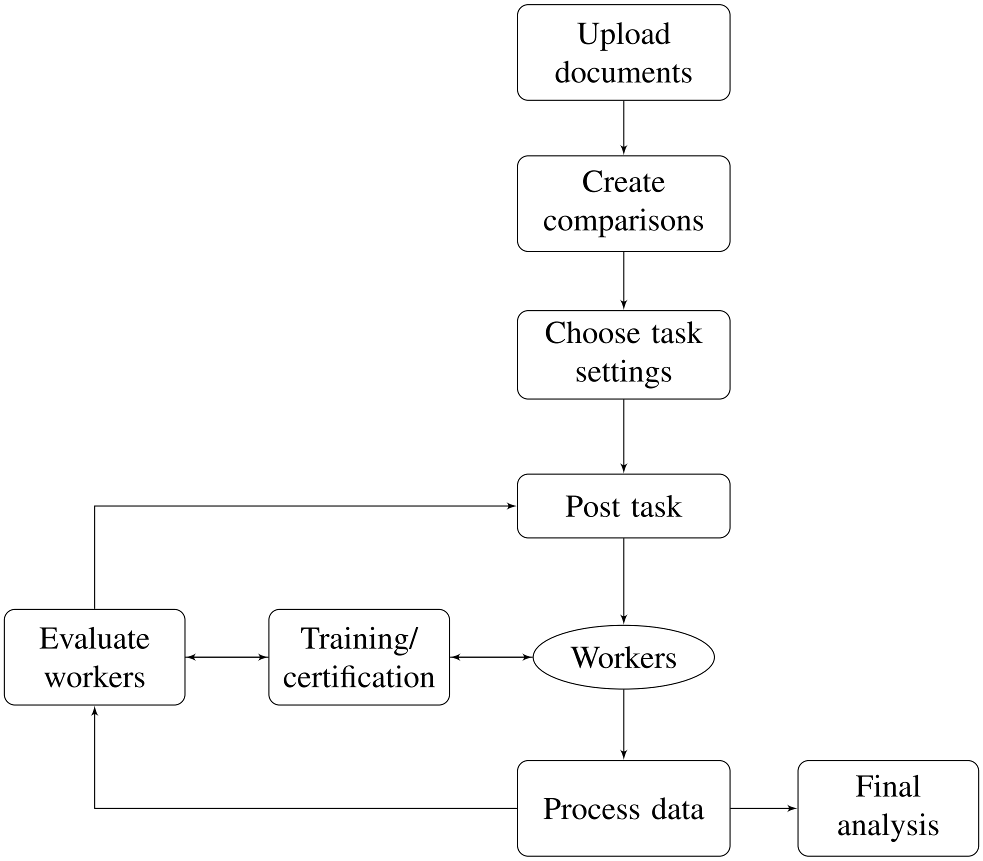
Finally, our pairwise comparison framework allows us to easily evaluate the data quality from individual workers as part of our statistical processing (discussed below). Rather than relying on attention filters (Berinsky, Margolis, and Sances 2014) or "gold standard" evaluations (Benoit et al. Forthcoming), we are able to assess the quality of the data from each worker on an ongoing basis as new data become available. `SentimentIt` provides easy-to-use tools for evaluating workers and removing certifications from workers providing low-quality data.

3.2. *Work flow*

The core functionality of the `SentimentIt` platform is a cloud-based web application that interacts smoothly with AMT to post jobs, certify workers, store responses, and generally reduce the difficulty for researchers wishing to utilize AMT workers. Researchers access the functionality

[6] Illustrative text from one training model is shown in Appendix SI-6. Our training modules were built in the Qualtrics survey software, which can interact with `SentimentIt` via application program interface (API) calls. Details for setting up this interface are provided in Appendix SI-7. Evidence on the effect of training on worker quality is provided in Appendix SI-8.

Figure 1: `SentimentIt` workflow



Note: The workflow is straightforward and can be flexibly altered at any point by the researcher. Certifications can be required and crafted to meet the needs of the specific task, and the researcher can evaluate data at any point to revoke certifications of low-quality workers. As much or as little of this workflow as desired can be automated based on the researcher's preferences (see Appendix SI-10).

of `SentimentIt` via application program interfaces (APIs) that can be called from any computing environment or platform (Python, Java, etc.). However, all of the functionality described below is fully integrated into our `R` package, making the process for researchers accustomed to the `R` language especially straightforward. The complete workflow is depicted in Figure 1.

First, we pre-process the textual data. This generally just involves ensuring our text is in a machine-readable format. However, where the documents are too long for simple comparison, we may choose to break the document into shorter, meaningful parts, such as paragraphs. The texts are then passed via API to `SentimentIt`. After the documents are in the system, we randomly pair the documents into a series of comparisons. For example, if we want 20 comparisons per document for 500 documents, we randomly create 5,000 unique comparisons ($500 \times 20/2 = 5,000$).[7] We then send paired document identification numbers and an associated question (e.g., "Which statement is more positive?") via API.

Once the comparisons are set up, they are ready to be sent to workers. At this stage, we can determine the task settings, dictating how much we want to pay workers and whether we require the worker to have a certification. We then send the comparisons out to the workers via an API call. In most cases, the complete universe of micro-tasks should not be posted simultaneously. We find that posting jobs in batches of 1,000 tasks allows us to keep track of how quickly the tasks are accomplished, and, importantly, gives us the opportunity to analyze the quality of the responses. If we determine that specific workers are providing poor data, we can revoke their certification via API and prevent them from further contaminating the data.

3.3. *Statistical modeling*

Once a sufficient number of the comparisons are complete, we can download the data via API. The data simply indicate which of the two documents was selected, the unique worker ID, and the time the task was completed. We then process the data using a random utility model, which creates document-level estimates along the dimension of interest (e.g., a measure of a document's positivity). Specifically, we model the probability that one document would be chosen over another, while estimating worker reliability given the choices made by that worker. Let i and j index documents in a comparison. Let k index the workers. The random utility model is specified as:

$$\Pr(y_{ijk} = j) = \frac{\exp(b_k(a_j - a_i)}{1 + \exp(b_k(a_j - a_i))} \tag{1}$$

The model is completed by specifying the following priors:

$$a_j \sim \mathcal{N}(0,1) \qquad b_k \sim tr\mathcal{N}(0,\sigma^2) \qquad \sigma \sim tr\mathcal{N}(0,3),$$

[7] Throughout the text, when we refer to random pairwise comparisons, we are randomly selecting comparisons from the set of all possible unique pairwise comparisons.

where $\mathcal{N}$ refers to the normal distribution, and $tr\mathcal{N}$ refers to the normal distribution truncated at zero to only support positive values.[8] We estimate the model using Hamiltonian Markov Chain Monte Carlo sampling using Stan (Carpenter et al. 2016).[9] In combination, this model produces posterior estimates for the documents' positions on the latent scale of interest (a_j) as well as the workers' reliability (b_k).

We can extend this model to allow a hierarchical structure. In our final application, we deal with large documents (State Department reports) that require simplification to create suitable micro-tasks. We therefore construct the pairwise comparisons using paragraphs rather than entire documents. To allow for a hierarchical structure of the data, let i and j still index paragraphs in a comparison and k index the worker. We now let m index the higher-level documents. The hierarchical random utility model is still specified as in Equation (1). However, the a estimates are now centered at a higher-level-document mean for document m, denoted θ_m, and the priors are,

$$a_j \sim \mathcal{N}(\theta_m, \sigma_m^2) \forall j \in m \qquad b_k \sim tr\mathcal{N}(0,1) \qquad \sigma_m \sim tr\mathcal{N}(0,.5) \qquad \theta_m \sim \mathcal{N}(0,1).$$

In both models,[10] the parameter estimates for the workers (b_k) give us an assessment of how well each worker performed. Intuitively, these estimates become lower for workers whose choices do not reflect how the documents are understood by other workers. If we estimate a worker as being a (low) outlier, we can ban the worker from future tasks. We find that revoking qualifications from these workers modestly improves the validity of our final estimates relative to benchmarks (see Appendix SI-5 for additional discussion of worker quality). Therefore, we recommend periodically running the statistical model and removing all workers who are obvious outliers. After

[8] Note that we set the variance term fro the prior of a_i and σ at 1 and 3 respectively to identify the scale of the latent distribution.

[9] The code for the complete Stan models used in our applications is shown in Appendix SI-3. Tuning parameters are set automatically through Stan.

[10] While there are many alternative ways of modeling this data, in our experience the resulting document-level estimates are largely invariante to these choices. For instance, Table SI-1 in Appendix SI-4 shows the correlations between the a_i estimates used in our applications below are correlated (Person's r) at $0.95 - 0.97$ with the arithmetic mean coder choice (where a document is codes as 0 when it is not chosen and 1 otherwise). Thus, while we feel that the model above is useful – particularly for identifying low-quality coders – the conclusions we draw below are robust to specific modeling choices and priors.

disqualifying problematic workers (if any), we can post further tasks and repeat as necessary.

Our R package can do as little or as much of this in an automated fashion as the researcher wants (see Appendix SI-10). The package can take in text and post the documents to SentimentIt, create comparisons, post tasks, check if the tasks are completed, download the data, test for worker outliers, ban unwanted workers, and repeat until all of the desired data have been collected and analyzed. If, instead, the researcher wishes to have more control of every stage, each step specified above can be controlled manually through R or via calls to the SentimentIt API.

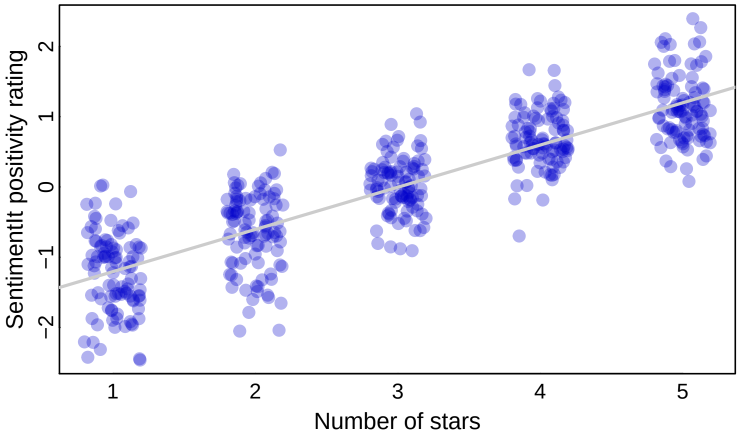
4 APPLICATIONS

Having described the details of our method, in this section we evaluate SentimentIt using texts from an online forum, a survey, political ads, and formal reports from a bureaucratic agency. In each exercise, we demonstrate that our estimates are valid measures of underlying latent traits of interest. To do this, we compare our estimates to relevant benchmarks created either by humans or automated methods. We show that our estimates correlate highly with these benchmarks and argue that where SentimentIt scores disagree with benchmarks, SentimentIt is actually better at capturing underlying latent traits. Further, for each example we demonstrate that the measures are reliable and replicable. Following the same procedure using the same settings in SentimentIt results in estimates that are highly correlated (Pearson's $r \geq 0.88$ in all cases).

4.1. *Movie reviews*

In our first application, we evaluate texts from an online forum. Specifically, we pulled 500 user-contributed movie reviews from Rotten Tomatoes along with their associated star ratings. Rotten Tomatoes allows users to rate movies on a five-star scale, and we selected 100 reviews from each category. We then apply the SentimentIt system to measure the positivity of each review using the text alone. We compare our measure to the star ratings chosen by the reviews' authors, which serve as a benchmark for validation (Pang and Lee 2005). While valuable as a benchmark, it is important to remember that star ratings are discrete and may have different meanings across individuals. Therefore, our first goal is to demonstrate that our estimates are strongly

Figure 2: `SentimentIt` movie review positivity estimates on number of stars



Note: As the number of user-provided stars increases, so do the estimates of positivity from `SentimentIt`. There is little overlap between estimates even as proximate as two stars.

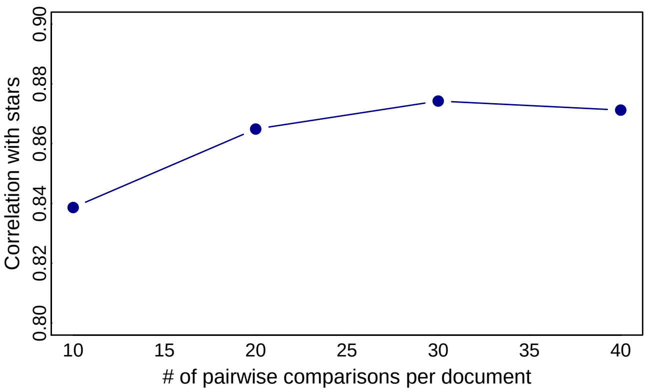
correlated with the star ratings. However, we also wish to show that where disagreement exists, the `SentimentIt` estimates better characterize the underlying text.

For this application, we created 40 random pairwise comparisons per document, resulting in 10,000 comparisons. We required participating workers to complete a qualification, and paid them $0.04 per task. We sent tasks to AMT in batches of 1,000 or 500 and analyzed the results between each batch to identify low-quality workers. Throughout the experiment we banned only two workers out of 126.

Figure 2 shows our estimates plotted against the number of stars assigned by the author.[11] The figure shows a very clear trend: as the stars increase so do our estimates (Pearson's $r = 0.87$). There is only a modest level of overlap between categories, with zero one-star ratings scoring higher than any five-star ratings. Further, Table 1 shows seven reviews where our positivity measure disagreed most with the user-provided star ratings. Our readings of these texts is that where `SentimentIt` estimates disagree with the stars, the star ratings seem to have been assigned in a manner not supported by the text. That is, we believe that our measure of the underlying sentiment more accurately reflects how a *standard* reader would translate the language into stars. For

[11] For the purposes of exposition, in the main text we focus exclusively on the point estimates (posterior means) for each document. However, the model also produces posterior measures of uncertainty for each estimate, which we discuss in Appendix SI-11.

Figure 3: Correlations between SentimentIt estimates and movie review stars



Note: Analyzing data after 10, 20, 30, and 40 comparisons shows that after 20 comparisons, the increase in correlations between SentimentIt estimates of positivity and movie review stars diminishes significantly.

instance, a review that reads (in part), "Almost plotless, but with moments that stick to your soul like a coating of grime," is language more consistent with a slightly negative review rather than the four-star rating assigned by the author.

How many comparisons are needed to generate valid estimates of latent traits embedded in texts? To provide an answer to this question, we estimated positivity measures using the first 10 pairwise comparisons (per document), the first 20 comparisons, the first 30, and then the complete dataset.[12] Figure 3 shows the correlation between these estimates of positivity and the user-provided stars. After ten comparisons per document, the correlation is 0.84. After twenty, the correlation is 0.86. After thirty and forty, the correlations are both about 0.87. Further, the point estimates after analyzing only 20 comparisons are virtually identical to the point estimates after analyzing 40 ($r = 0.99$). There is a very mild gain in precision, with the mean standard deviation of the estimates decreasing from 0.26 to 0.22 as we move from 20 to 40 comparisons. Thus, in this experiment there is little benefit to adding tasks beyond 20 comparisons.

We also wished to estimate the reliability of our measure. Figure 4 plots the point estimates generated by analyzing *only* the first 20 comparisons and those estimated using *only* the last 20 comparisons. The correlation between these estimates is 0.92. This is strong evidence that the SentimentIt measures are highly reliable and that 20 comparisons is an adequate number to

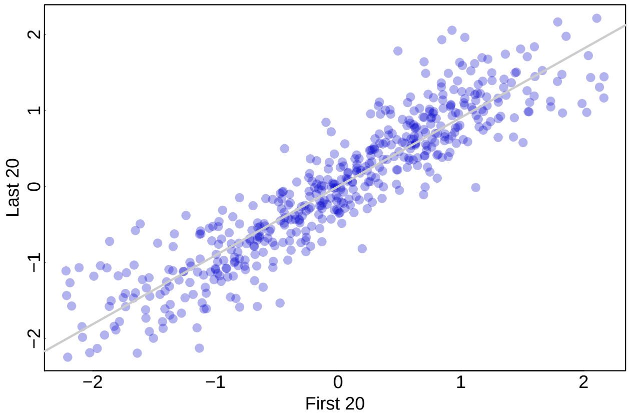
[12] Pairwise comparisons were constructed in blocks of 10 to make this analysis feasible.

Table 1: Reviews with largest disagreement between SentimentIt scores and star ratings

Review text	Positivity score	Stars
“I really enjoyed the original with Brooke Shields and yes it is on my movies that are very hard to find list, however deemed this sequel just ok. Somethings they should just leave as they are.”	0.03	1
“Eddie Murphy. What happened to your taste in movie roles? This was the stupidest movie..not funny at all. Just stupid.”	-2.05	2
“A film that quite possibly showcased Monroe playing herself, especially late in her career. Nell was unstable and not able to handle the pressures of the outside world (hence the metaphor of being a sort of live-in nanny, cut off from the rest of the world but protected as well). Monroe’s’s late-career personal troubles and demons were well documented and attested by her suicide. Even though this film was early on in her career it offers a chilling and eerie view of what’s to come for the legendary actress. They say the best of the best actors and actresses draw upon their very souls to come up with their startling performances. I think Monroe was pulling from her very core in this film. Worth a viewing and analysis.”	1.04	3
“I think Buster Keaton is one of the more inconsistent actors from the silent film era. I really didn’t like The general but I adored Sherlock Jr. This one I would say is ok. Buster and the woman who rejected to marry him accidentally both end up on a ship at sea alone. In this journey they encounter a storm, cannibals, and a scary painting of sailer. Now there were some nice laughs in here but at the same time for a film only an hour long I shouldn’t have been bored as much as I did.”	-0.91	3
“Residents of an institution escape and wreck the grounds with childlike acts of vandalism and petty cruelty. The entire cast is composed of dwarfs. Almost plotless, but with moments that stick to your soul like a coating of grime (tiny Hombre laughing at the struggling camel may haunt your nightmares for years to come). Animal lovers beware.”	-0.70	4
“The best sequel and best boxing film since Rocky. Stallone is superb and now I know why he was applauded and Michael B. Jordan is amazing as man trying to respect his legacy but find his own path to victory. Also good is Thomson as Jordan’s singer love interest. Great directing, writing and cinematography. Was cheering in my seat and left the theatre feeling satisfied.”	1.67	4
“This is a story of an unjust system, looking only to protect its own neck. This is the outrage of onlookers and commentators who cannot stand the ridiculous logic behind taking a man’s life away when he did not commit any wrong. Most of all, this is one young man’s brave fight to show the world that he can be beaten down, but not beaten. The story is reminiscent of Mumia Abu Jamal’s own plight, except the circumstances of Paco’s innocence are more blatant and apparent.”	0.08	5

Note: It is evident that when SentimentIt estimates movie reviews in an unexpected way given the star ratings, the star ratings seem to be given in a way that differs from average ratings.

Figure 4: Reliability of `SentimentIt` measures of positivity in online movie reviews



Note: We analyzed the first 20 and last 20 pairwise comparisons estimating the positivity of movie reviews separately. This plot shows the high correlation (0.92) and therefore high reliability between runs.

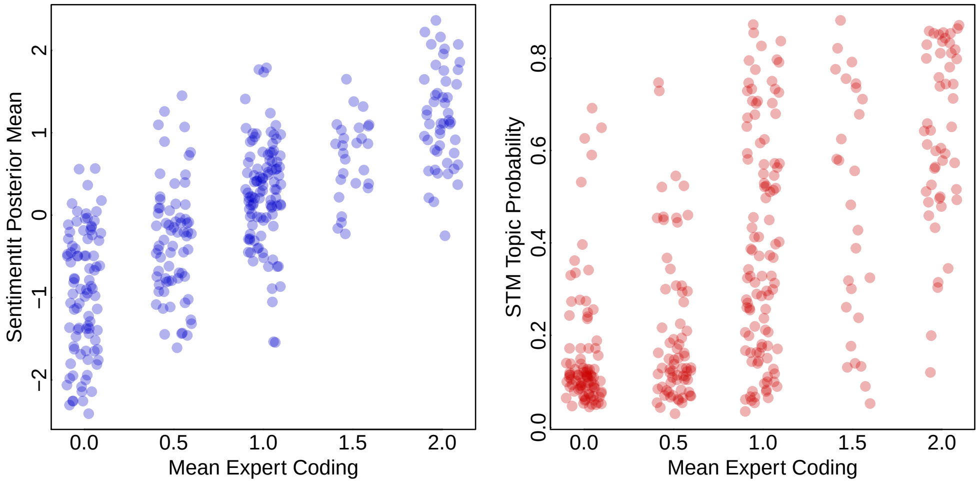
minimize costs while maximizing reliability.

Finally, in Appendix SI-1, we provide evidence from additional analyses of online movie reviews that further illustrate these points. Importantly, in this application, data was collected over a month apart using mostly different workers, yet the reliability estimates are virtually the same. In Appendix SI-2, we benchmark `SentimentIt` against an advanced supervised learning method (Socher et al. 2013) using a similar dataset to demonstrate that our approach surpasses even the best available automated methods for supervised sentiment analysis.

4.2. *Open-ended survey responses*

In our next application, we analyze a corpus of open-ended survey responses about immigrants. Specifically, we analyze statements collected by Gadarian and Albertson (2014, p. 139), who asked respondents on a national survey, “When you think about immigration, what do you think of?” This example is useful because these statements have been analyzed using both human content coding and an automated structural topic model (STM). Gadarian and Albertson (2014) used two trained research assistants to code each statement as indicating no (0), some (1), or extreme (2) levels of anxiety towards immigrants or immigration. They averaged the two coders’ evaluations to generate a 2-point scale for each response. Roberts et al. (2014) subsequently analyzed these same statements using a structural topic model (STM), identifying one topic as indicating fearfulness.

Figure 5: Fearfulness estimates from `SentimentIt` and STM on expert coding



Note: The fearfulness estimates from `SentimentIt` increase as do the human codes. The STM estimates, however, tend to pool towards zero and perform particularly poorly at higher human codes.

Thus, we are able to assess `SentimentIt` relative to both alternative approaches.

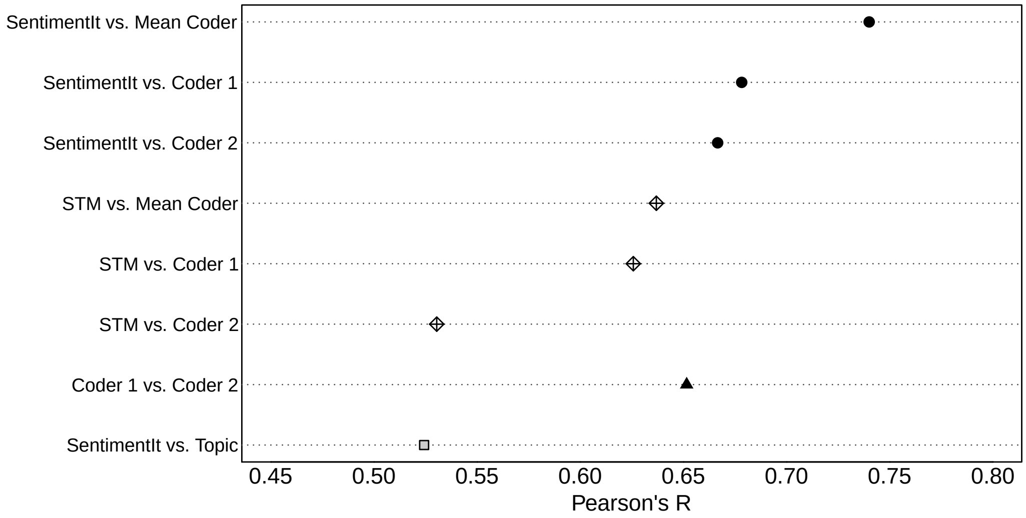
We analyze the survey responses with 20 pairwise comparisons per document. We required a certification and paid $0.04 per task. Workers were instructed to choose which statement indicated the greater degree of fear towards immigrants.[13] The entire procedure was completed in one working day. In total, 84 workers participated and we banned none.

The left panel of Figure 5 compares the `SentimentIt` estimates relative to the mean expert-coder rating. The right panel shows how the mean coder rating compares to the probability that a document is assigned to the "fearfulness" topic by the STM model (Roberts et al. 2014). The figure shows that that as expert coding increases, our measures do as well. However, the STM does not seem to be capturing this sense of fearfulness as accurately. Specifically, the STM model tends to pool towards the low end, and the predictive validity of the STM model is particularly poor at capturing moderate and high levels of fearfulness.

[13] The full prompt was:

> In a survey, researchers asked respondents, "When you think about immigration, what do you think of?" Below are two answers to that question. Please indicate which of the two statements expresses more fear of immigrants or immigration. Remember, we are not interested in whether the writer has a positive or negative view of immigrants. We are only interested in whether the writer is expressing fear of immigrants or immigration.

Figure 6: Correlations between different sources of fearfulness estimates



Note: The correlations between SentimentIt and the human coders are notably higher than the correlations between STM and the coders. The human coders are also poorly correlated.

Figure 6 compares the correlations between SentimentIt, each expert coder, the mean of the expert codes, and the STM topic probabilities. As expected, the SentimentIt measures correlate with the expert coding much better than does STM. SentimentIt is correlated with the mean expert rating at 0.74 while STM is only correlated at 0.64. Indeed, the SentimentIt measures correlate more highly with each expert coder than the individual coder scores correlate with each other.

Table 2 shows the largest errors for both STM and SentimentIt when the human coders unanimously determined that responses indicated no anxiety or extreme anxiety. The first four entries are the responses that each automated method identified as being fearful where the human coders unanimously identified them as non-fearful. As can be seen, the topic model estimates several benign responses as fearful, such as “nothing” and “illegallanguage” (sic). SentimentIt, however, disagrees with the human coders on responses that exhibit relatively strong language and anxiety about illegal immigration (e.g., “learn the damn langwige(sp) and learn our trafic laws.”).

The bottom three entries are statements that the automated methods identified as non-fearful where the expert coders unanimously identified them as extremely fearful statements. We see that the topic model categorizes some responses as low that actually exhibit strongly fearful emotions mentioning things such as “crime”, “trouble”, and “changing the basic makeup of the United States.” `SentimentIt`, on the other hand, seems to be providing better estimates than even the expert coders. That is, the statements estimated as minimally fearful by `SentimentIt` but extremely fearful by the expert coders contain mostly banal statements about policies that are not obviously different from the statements coded unanimously non-fearful in the first four rows.

Finally, we turn to the question of reliability. Note that the two expert coders provide estimates that are correlated at only 0.65. Further, although in a trivial sense the topic model is perfectly reliable,[14] in practice the model estimates are a subjective choice of the researcher since the topics identified by the algorithm change significantly depending on small changes in initial conditions. So, for instance, Roberts et al. (2014) ran 50 topic models to analyze this dataset and chose one “based on exclusivity and semantic coherence criterion” (p. 1073).[15] In comparison, the `SentimentIt` procedure is transparent, reliable, and does not rest on the judgement of the researcher. Approximately two weeks after our initial data collection, we exactly duplicated our process and ran 20 more comparisons with the same settings and analyzed the resulting data separately. Figure 7 shows that the correlation between these two `SentimentIt` analyses is a very impressive $r = 0.89$.

4.3. *Congressional advertisements*

For several election cycles, the University of Wisconsin Advertising Project (WiscAds) collected and coded political ads in the United States and released it for political science research. In this application, we focus on the “tone” of ads, or their level of negativity, which is a measure widely used in political science research (e.g., Freedman and Goldstein 1999; Goldstein and Freedman 2002). In the WiscAd dataset, ad tone is determined by expert coders who categorized ads as either promoting a single candidate, contrasting two candidates, or attacking a candidate. If the ad is contrasting, it is further categorized as either being more aimed at promoting than attacking,

[14] Starting the algorithm using the same random seed will necessarily lead to the same estimates.

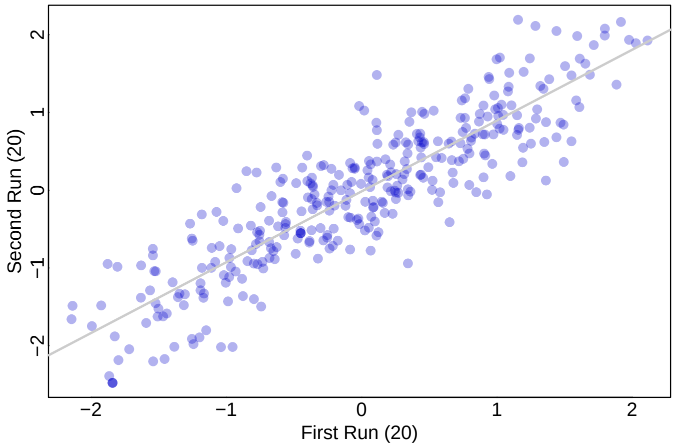
[15] A technical solution for the sensitivity of structural topic models to starting values is addressed in subsequent work (Roberts, Stewart, and Tingley 2016).

Table 2: Largest differences among unanimous "none" and "extreme" statements

Mean expert	Topic model	STM prob.	`SentimentIt`	Estimate
None (0)	"nothing"	0.65	"that the ileagles who sneack over the border should be fined to the fullest extent. i worked most of my life for what little i get. i on't think it's right for them to receive everything free."	0.36
None (0)	"they do not pay taxes"	0.69	"i think if they come in to the country ileaglely they should be depored . do it the right way or stay the hell home and if you do get here the correct way learn the damn langwige(sp) and learn our trafic laws."	0.56
None (0)	"illegallanguage"	0.63	"they get all our stuff for free, and it isnt right"	0.17
None (0)	"nothing"	0.59	"we need to seal the borders. both north and south. fine all the employers that hire illegals and also those who rent to them. if they can't find an job nor a place to live they will go home."	0.57
Extreme (2)	"changing the basic makeup of the united states. creating a population with more socialistic values. mexico is largely an economic failure, so the immigrants from there may tend to have the values that created that failure."	0.20	"the cost of living, cost of food, diseases, and jobs that are taken from americans."	0.37
Extreme (2)	"too many non speaking english americans. too many people that do not stand with what we were founded on. too much trouble wanting their ways and not becoming americanized"	0.12	"schools, hospitals, total infrastructure of our communities"	−0.25
Extreme (2)	"bad economy, crime rates, unemployment, money under the table..."	0.16	"social security health coverage job security poverty level"	−0.21

Note: It is evident that when `SentimentIt` estimates differ from human coded statements, the text appears to be more in line with `SentimentIt` scores. When the structural topic model misclassifies, however, the topic model is more likely to be an error than the human coders.

Figure 7: SentimentIt fearfulness estimates from separate runs



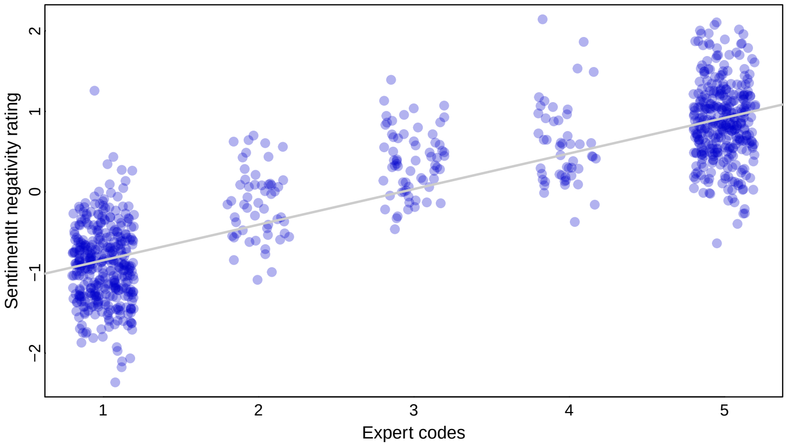
Note: We ran two rounds of twenty comparisons each on the fearfulness of survey responses regarding immigration two weeks apart. This plot shows the high correlation (0.89) and therefore high reliability between independent runs.

more attacking than promoting, or equally attacking and promoting. The result is a five-point scale of negativity ranging from one (positive) to five (attack). We apply SentimentIt to analyze all televised ads for the U.S. Senate in 2008 ($n = 942$) and compare our estimates to this five-point negativity scale. This allows us to again validate SentimentIt against a meaningful benchmark. Further, this example illustrates how a tedious expert-coding task can be more easily accomplished via the automated SentimentIt approach while actually improving the reliability and validity of the measure.

For each ad, we created 20 pairwise comparisons for a total of 9,420 tasks. Workers were required to complete an extensive training module and were paid $0.06 per comparison. Workers were instructed to select the ad that was, "most negative towards the candidate(s) mentioned, or least positive about the candidate(s) mentioned." In all, 123 workers participated in the task and none were banned.

Figure 8 shows our estimates plotted against the five-point negativity scale. The SentimentIt scores are correlated with the expert codings at $r = 0.85$. This is a very high correlation, but there are some disagreements. Table 3 shows a few of the greater disagreements between the SenitmentIt measures and those generated by the expert coders. The first example is coded as positive by the expert coders, but the ad is negative in tone and appears to be a strongly neg-

Figure 8: `SentimentIt` negativity rating relative to five-point expert codes



Note: Using the five-point scale of human codes, we can see that as human codes increase, so do our estimates. In general, the differentiation between codes is evident. The largest disagreements are happening in the middle categories, discussed in the text.

ative contrasting ad that was mis-coded. The second ad is coded as contrast (2 = more positive than attack), but `SentimentIt` estimates it as having almost no negative content. In fact, the ad does not mention an opposing candidate or party and is another clear mis-coding. The third ad is coded as contrasting by expert coders because it mentions both candidates by name. However, the ad mostly consists of strong and negative language, causing our estimate of negativity to be quite high (1.39). Finally, the last example is coded as an attack by human coders because it does not provide positive information about a named candidate. However, the ad only mentions the target of the attack (Collins) in one sentence and otherwise conveys positive information about the Employee Free Choice Act. These last examples illustrate that the strict coding rules designed to improve the reliability of content analysis can obscure the underlying latent trait of interest.

To determine the reliability of our measure, we repeated the exercise 35 days later with 20 more comparisons on a random sample of half the ads ($n = 467$) using the same procedure as before. In all, 52 workers participated in this exercise. We revoked the certification from only one worker. The correlation between the two runs is 0.90.

4.4. *Measuring torture using human rights reports*

In our final application, we demonstrate how `SentimentIt` can be generalized to larger documents. Specifically, we turn to the sobering task of coding the section entitled, “Torture

Table 3: Example ads where SentimentIt disagreed with expert coders

Advertisement	SentimentIt	Experts
[Priscilla Lord Faris]: "Early in this campaign I believed that Al Franken could defeat Norm Coleman. But, no matter how many millions he spends it is clear that his history of pornography, degrading women and minorities, and his questionable financial transactions will continue to be the focus of blistering Republican attack ads. I represent real Minnesota values as a mother, a teacher, a volunteer, and an advocate. I'm Priscilla Lord Faris, I approve this message, and ask for your vote September 9th."	1.26	Positive (1)
[Steve Novick]: "I'm Steve Novick and I approve this message." [John Kitzhaber]: "I'm John Kitzhaber and I approve of Steve Novick. Negative politics as usual or something different? Steve Novick is not a typical politician and he's not running a typical campaign. Steve is standing up for principle and that's why Oregon Democrats are standing up for Steve. Oregon teachers are supporting Steve, so are papers across the state. And I think Steve Novick is the only candidate we can count on for real healthcare reform. Steve Novick, the cure for politics as usual."	-1.09	Contrast (2)
[Announcers]: This isn't complicated. Roger Wicker serves with honor and integrity. Ronnie Musgrove. His ethics? Shameful. Roger Wicker. Supported by Thad Cochran, the VFW, the NRA. Ronnie Musgrove. Supported by pro-abortion, pro-gay marriage groups. Roger Wicker. Never voted for a pay raise, always supports Social Security. Ronnie Musgrove. Failed governor, lost jobs, the beef plant scandal and now he's lying about Roger Wicker. This isn't complicated. Roger Wicker. [Roger Wicker]: "I'm Roger Wicker, and I approve this message."	1.39	Contrast (3)
[Announcer]: CEO's salaries and benefits are getting fatter and fatter... while workers face soaring gas prices, foreclosures, and rising healthcare costs. The Employee Free Choice Act gives workers the freedom to form a union so they can earn better wages, retirement security, and healthcare coverage. Call Senator Susan Collins tell her to support the Employee Free Choice Act and stop siding with wealthy CEO's over working families. American Rights At Work is responsible for the content of this advertising.	-0.64	Attack (5)

Note: The first two examples show errors in expert coding, as the ads were put in the wrong category. In the latter two, it is evident that when SentimentIt estimates differ from human codes, expert coding scheme's reliance on strict coding rules mischaracterizes the overall tone within the ads.

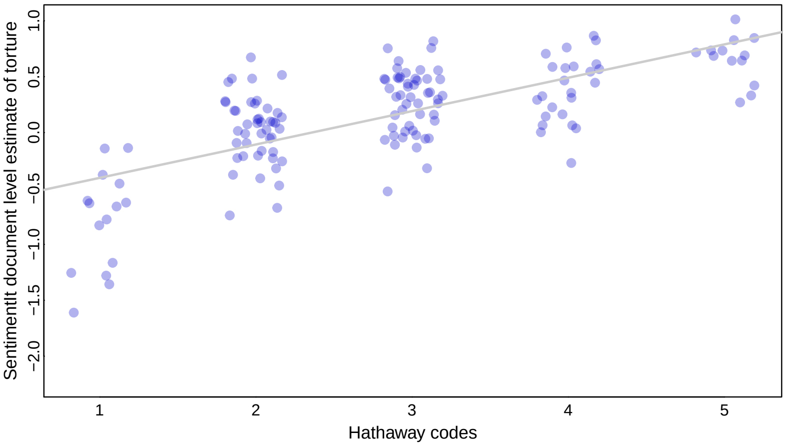
and Other Cruel, Inhuman or Degrading Treatment or Punishment," from all of the U.S. State Department Human Rights Reports issued annually for nearly every country in the world (Fariss et al. 2015). In this application, we use reports from 1999 to create a continuous scale indicating the amount of torture conveyed in the report. Hathaway (2002) codes these documents by hand on a five-point scale, with higher values indicating more entrenched, brutal, or frequent torture. Using this measure, Hathaway (2002) argues that ratifying human rights treaties is associated with greater degrees of torture (see also Neumayer 2005). We emphasize that our aim is not to capture the actual amount or severity of torture in each country – a task well beyond the scope of this article – but only to measure the concept of torture as it is expressed in these reports.

Since many of these reports are quite lengthy and would be difficult for even an expert to read and meaningfully compare, we divided the 182 documents into 1,652 paragraphs. We created 20 comparisons per paragraph. Since reading and understanding these documents is far more challenging than the examples above, we paid workers $0.10 per task and required an extensive certification training (see Appendix SI-6). We asked workers to indicate which paragraph suggested more torture, which was defined in detail during the training. Our definition and coding rules were designed to approximate those provided by Hathaway.

We posted the tasks in batches of 1,000. The process of gathering the data took approximately three days, and 81 workers completed tasks (we banned none). To analyze the data, we adjusted our statistical model to allow for a hierarchical structure (paragraphs within documents) as described in Section 3. The resulting document-level estimates are correlated with Hathaway's coding at 0.69. Figure 9 plots the `SentimentIt` estimates of torture to the Hathaway code. In Figure 10, we show a subset of the document- and paragraph-level estimates generated by our procedure.

Broadly speaking, our measure is consonant with Hathaway's coding. For instance, all of our estimates for the countries coded as one (no torture) by Hathaway fall within the first quantile of our estimates. Yet, there are significant disagreements. For instance, among those cases coded as a one, our estimates are noticeably higher for specific cases. Our estimate for Mali (-0.14) is the highest among all countries coded as one in Hathaway's scheme. Two paragraphs drive this result:

Figure 9: `SentimentIt` document-level estimates of torture on Hathaway codes

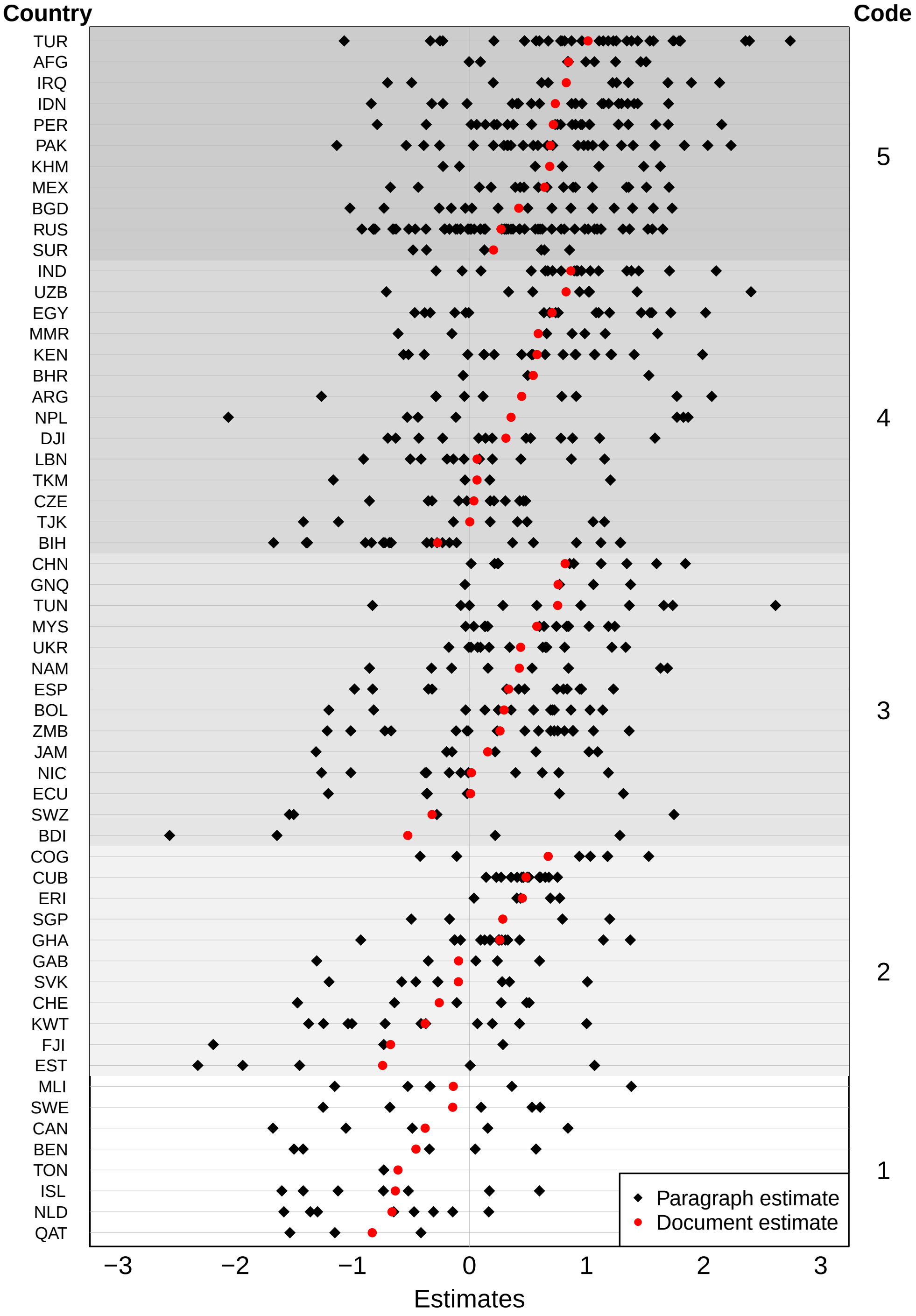


one explains at length the harsh prison conditions of the country (estimated at 0.36) and the other explicitly states Amnesty International reported that, “security forces had tortured ... in order to extract confessions” (estimated at 1.38). On the other hand, we estimate some countries coded as two slightly lower than expected. We estimate Estonia lower than any other country in category two (-0.74). The only evidence of torture from this document is a paragraph stating, “police used excessive force and verbal abuse during the arrest and questioning of suspects,” and “[p]unishment cells ... continued to be used” (estimated at 1.07).

The Republic of the Congo is coded as a two by Hathaway, but our estimate (0.67) is higher than any other document coded as a two. The report states explicitly that police and security forces were using beatings, rape, unwarranted strip searches, and unlawful imprisonment to solicit confessions, impose power, and punish. Our estimate for Cuba is also relatively high (0.48), which is also coded as a two by Hathaway. The Cuba document is a lengthy account of many acts of violence by police, unwarranted detention, acts of indirect violence against those who do not support the government, and harsh prison conditions.

Burundi is coded as a three in Hathaway, but we estimate Burundi to be on the lower end (-0.53). The first paragraph reads “members of the security forces continued to torture and otherwise abuse persons. In one such case, [Amnesty International] reported that members of the security forces were believed to have withheld food from detainees and beaten one of them severely. There were no known prosecutions of members of the security forces for these abuses” (estimated

Figure 10: `SentimentIt` document- and paragraph-level estimates of randomly selected countries and countries estimated differently than Hathaway



Note: Document-level estimates are shown in red, and paragraph-level estimates are shown in black. There is a clear trend that as Hathaway's codes increase, so do ours. The notable exceptions are explicitly included and discussed in the text. Where our estimates deviate from Hathaway's codes, there is strong reason to believe estimates from `SentimentIt` are outperforming expert codings.

at 1.29). This is a clear indicator of torture, but it is the only mention of torture in the document making it not obviously distinguishable from other reports coded as two in the Hathaway scheme. On the other hand, China (0.82), Tunisia (0.75), and Equatorial Guinea (0.76) are all categorized as threes, but we estimate rather high degrees of torture for these countries. The documents for all three contain multiple paragraphs detailing severe torture including beatings, administering electric shocks, hanging prisoners and suspects by their wrists, and torturing detainees to death.

Bosnia and Herzegovina is coded as a four, but we actually estimate it at slightly lower than the mean amount of torture (-0.27). The document has clear instances of violence, but overwhelmingly the violence described is not at all related to police or government acts of torture. Some of the paragraphs are actually positive, for example: "[i]nternational community representatives were given widespread and for the most part unhindered access to detention facilities and prisoners in the RS as well as in the Federation" (estimated at -1.39). Though this was clearly a tumultuous time for Bosnia, very little of the document speaks to torture. The same is true for Turkmenistan (0.06), Tajikistan (0.003), Lebanon (0.07), and the Czech Republic (0.04). The overall tone of all these documents is not very negative, at least with regard to torture.[16]

Based on these and other examples, we believe that the `SentimentIt` coding more faithfully reflects the level of torture in these documents than the original human-coded measures. However, to further assess the validity of these estimates, we compare our measure with three other well-known measures of torture drawn (in part) from this same document set. Specifically, we analyze the State Department variable of the Political Terror Scale (PTS) data on human rights violations (Gibney et al. 2015), the Ill-Treatment and Torture (ITT) data (Conrad and Moore 2012), and the torture variable from The CIRI Human Rights Dataset (Cingranelli, Richards, and Clay 2014).[17] We also test whether ratification of a torture convention is positively correlated with the amount of torture in a country as hypothesized by Hathaway. Hathaway (2002) finds that signing an anti-torture convention is positively but unreliably associated with torture. For each variable,

[16] Space does not a permit a fuller discussion of all the ways in which our coding disagrees with the Hathaway scheme. However, the full document set along with our estimates will be included in the replication archive for this article at the time of publication.

[17] Additional information on these measures are provided in Appendix SI-9.

Table 4: Standardized OLS coefficients for SentimentIt and Hathaway measures regressed on torture measures and treaty ratification

	Dependent variable:							
	ITT		PTS		CIRI		Torture Convention	
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
Hathaway	**0.724**		**0.263**		**0.275**		0.252	
	(0.210)		(0.079)		(0.064)		(0.523)	
SentimentIt		**0.964**		**0.298**		**0.275**		**1.296**
		(0.205)		(0.078)		(0.064)		(0.517)
Observations	93	94	106	108	103	105	106	108
R^2	0.382	0.442	0.654	0.663	0.511	0.510	0.343	0.367

Note: Bolded coefficients are significant at the $p \leq 0.05$ level. Additional controls for per capita GDP, population, population growth, trade, foreign aid, GDP growth, civil war, level of democracy, and nation durability not shown. SentimentIt and Hathaway codes are standardized for comparability. CIRI increases as the degree of torture decreases, and is reverse coded in the analyses.

we conduct separate regressions and calculate standardized regression coefficients. We maintain all controls of the original analysis. Our only deviation is that we look at only one year while the original analysis included 15 years of reports.

In almost all cases, we find that both the Hathaway and SentimentIt scores are positively related to the other measures ($p \leq 0.05$). The only exception is that the relationship between level of torture and ratifying an anti-torture convention is estimated unreliably using Hathaway's coding. However, although both measures are correlated with the alternative torture measures, both the standardized coefficients and R^2 values indicate that the SentimentIt measure is more highly related to these other variables. The only exception is the CIRI measure, where the two approaches are essentially indistinguishable in terms of their predictive validity.

Finally, to test for reliability of our measures we separately estimate the (roughly)[18] first ten comparisons to the last ten comparisons for each paragraph. The correlation between paragraph estimates is 0.77. The correlation between document estimates is 0.88. Considering we are only comparing the estimates between two rounds of ten comparisons, we consider this to be strong evidence indicating the SentimentIt platform is generating highly reliable measures.[19]

[18] Because the number of paragraphs was not divisible by 1,000, our last batch only had a few hundred comparisons, necessitating the division into halves be slightly less than perfect.

[19] Further, this level of reliability is actually higher than the expert coding procedure reported by Hathaway (2002,

5 CONCLUSION

Human language is central to the processes and outcomes of politics. However, this rich source of data is often overlooked in favor of more easily quantified behaviors. Many studies that do utilize texts and speeches rely on hand coding by trained experts, which is time-consuming, costly, and often unreliable. More recently, scholars have turned to automated text analysis algorithms that, while very promising, can perform poorly when uncovering the underlying latent sentiment of political texts and are primarily aimed at classification rather than measuring continuous traits.

In this paper, we propose a novel human computation framework approach to analyzing political text and recovering latent sentiment that provides the reliability of automated methods but leverages the superior abilities of humans to read and understand natural language. Specifically, we rely on having an online workforce complete pairwise comparisons of texts. By having the workers indicate which of two documents is more extreme along a dimension of interest (e.g., positivity), including documents in multiple pairwise comparisons, training and monitoring workers, and statistically post-processing the worker evaluations, we are able to reliably produce valid estimates of latent traits within texts. Further, we provide software that can automate as little or as much of the process as the researcher desires and have provided details about the workflow and steps needed to replicate our approach.

To demonstrate the benefits of the `SentimentIt` system, we apply it to measure positivity in movie reviews, fearfulness in open-ended survey responses about immigration, negativity of political ads, and levels of torture indicated in U.S. State Department human rights reports. In each application, we use meaningful benchmarks to show that estimates produced by `SentimentIt` are reliable and valid measures of underlying latent qualities. We believe that these results show that our approach can be fruitfully applied to the analysis of natural language in a wide variety of applications across subfields in political science and beyond.

Though the method we propose is quite powerful, it is not entirely unproblematic. First, although the examples above show the versatility of the method, there may be unknown limits its

p. 1971) (Cohen's κ=0.8).

applicability. Asking coders to evaluate the "tone" of a political ad may differ in kind from asking them to evaluate the quality of the legal reasoning in court cases. Moreover, the tasks we designed were specifically aimed at evaluating aspects of a text that are somewhat objective. Thus, bias in the pool of workers – even if widely shared – is unlikely to contaminate the data. However, asking workers to draw more deeply on their own judgement to evaluate, say, the "persuasiveness" of a specific text may require a more representative workforce. Whether the `SentimentIt` system can be successfully applied in these situations requires further investigation.

However, the most obvious limitation is that, though the system is much cheaper than human coding, it is not free. If we are coding medium-sized document sets, this is a relatively trivial problem. For example, our second application involving open-ended survey response cost approximately $130 and a few hours to complete. Indeed, we believe that `SentimentIt` may generally be more cost effective than relying on expert coders in almost all cases. However, the method is clearly infeasible with large document sets containing tens of thousands of documents and is much more expensive than relying on unsupervised automated methods.

Nonetheless, integrating the evaluations of online workforces into the process of encoding text is not limited to the exact procedure we discussed above. Moving forward, we plan to use this platform to handle larger document sets by further melding together the power of human workers and computer algorithms. First, we can use `SentimentIt` to build supervised learners by providing high-quality training sets. Second, we plan to extend the method to be more dynamic, combining supervised machine learning algorithms with pairwise comparisons from online workforces dynamically to allow sophisticated algorithms to *themselves* post comparisons to gain more information about specific documents. Finally, the platform could also be used as a verification of existing automated methods. That is, when researchers use a text-analysis algorithm, they can analyze some documents to assess the validity of the results (Grimmer and King 2011).

5 References

Alonso, Omar, and Ricardo Baeza-Yates. 2011. *In advances in information retrieval.* Springer chapter Design and implementation of relevance assessments using crowdsourcing.

Alonso, Omar, and Stefano Mizzaro. 2012. "Using crowdsourcing for TREC relevance assessment." *Information Processing and Management: an International Journal* 48(6): 1053–1066.

Baumgartner, Frank R., and Bryan D. Jones. 1993. *Agendas and instability in American politics.* Chicago: University of Chicago Press.

Benoit, Kenneth, Drew Conway, Benjamin E. Lauderdale, Michael Laver, and Slava Mikhaylov. Forthcoming. "Crowd-sourced text analysis: Reproducible and agile production of political data." *American Political Science Review* .

Berinsky, Adam J., Gergory A. Huber, and Gabriel S. Lenz. 2012. "Evaluating online labor markets for experimental research: Amazon.com's Mechanical Turk." *Political Analysis* 20(3): 329–50.

Berinsky, Adam J, Michele F Margolis, and Michael W Sances. 2014. "Separating the shirkers from the workers? Making sure respondents pay attention on self-administered surveys." *American Journal of Political Science* 58(3): 739–753.

Bohannon, John. 2011. "Social science for pennies." *Science* 334(6054): 307–307.

Brady, Henry E. 1985. "The perils of survey research: Inter-personally incomparable responses." *Political Methodology* pp. 269–291.

Budge, Ian. 2001. *Mapping policy preferences: Estimates for parties, electors, and governments, 1945-1998.* Vol. 1 Oxford University Press.

Budge, Ian, and Paul Pennings. 2007. "Do they work? Validating computerised word frequency estimates against policy series." *Electoral Studies* 26(1): 121–129.

Callison-Burch, Chris, and Mark Dredze. 2010. Creating speech and language data with Amazon's Mechanical Turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk.* Association for Computational Linguistics pp. 1–12.

Carpenter, Bob, Andrew Gelman, Matt Hoffman, Daniel Lee, Ben Goodrich, Michael A. Betancourt, Michael Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. 2016. "Stan: A probabilistic programming language.".

Cingranelli, David L., David L. Richards, and K. Chad Clay. 2014. "The CIRI Human Rights Dataset.".

Conrad, Courtenay R., and Will H. Moore. 2012. "Ill-Treatment and Torture (ITT) Dataset.".

Druckman, James N., Martin J. Kifer, and Michael Parkin. 2009. "Campaign communications in US congressional elections." *American Political Science Review* 103(3): 343–366.

Fariss, Christopher J, Fridolin J Linder, Zachary M Jones, Charles D Crabtree, Megan A Biek, Ana-Sophia M Ross, Taranamol Kaur, and Michael Tsai. 2015. "Human rights texts: converting human rights primary source documents into data." *PloS one* 10(9): e0138935.

Fenno, Richard F. 1978. *Home style: House members in their districts.* Boston: Little, Brown.

Freedman, Paul, and Ken Goldstein. 1999. "Measuring media exposure and the effects of negative campaign ads." *American journal of political Science* pp. 1189–1208.

Gadarian, Shana Kushner, and Bethany Albertson. 2014. "Anxiety, immigration, and the search for information." *Political Psychology* 35(2): 133–164.

Gibney, Mark, Linda Cornett, Reed Wood, Peter Haschke, and Daniel Arnon. 2015. "The Political Terror Scale 1976-2015.".

Goldstein, Ken, and Paul Freedman. 2002. "Campaign advertising and voter turnout: New evidence for a stimulation effect." *Journal of Politics* 64(3): 721–740.

Grimmer, Justin. 2010. "A Bayesian hierarchical topic model for political texts: Measuring expressed agendas in Senate press releases." *Political Analysis* 18(1): 1–35.

Grimmer, Justin. 2013. "Appropriators not position takers: The distorting effects of electoral incentives on congressional representation." *American Journal of Political Science* 57(3): 624–642.

Grimmer, Justin, and Brandon M Stewart. 2013. "Text as data: The promise and pitfalls of automatic content analysis methods for political texts." *Political Analysis* p. mps028.

Grimmer, Justin, and Gary King. 2011. "General purpose computer-assisted clustering and conceptualization." *Proceedings of the National Academy of Sciences* 108(7): 2643–2650.

Hathaway, Oona A. 2002. "Do human rights treaties make a difference?" *The Yale Law Journal* 111(8): 1935–2042.

Henderson, John A. 2015. "Using experiments to improve ideal point estimation in text with an application to political ads.".

Hitler, Adolf. 2013. *Hitler's second book: the unpublished sequel to Mein Kampf*. New York, NY: Enigma Books.

Hopkins, Daniel J, and Gary King. 2010. "A method of automated nonparametric content analysis for social science." *American Journal of Political Science* 54(1): 229–247.

Hsueh, Pei-Yun, Prem Melville, and Vikas Sindhwani. 2009. Data quality from crowdsourcing: a study of annotation selection criteria. In *Proceedings of the NAACL HLT 2009 workshop on active learning for natural language processing*. Association for Computational Linguistics pp. 27–35.

Ipeirotis, Panagiotis G, Foster Provost, Victor S Sheng, and Jing Wang. 2014. "Repeated labeling using multiple noisy labelers." *Data Mining and Knowledge Discovery* 28(2): 402–441.

Jamal, Amaney A., Robert O. Keohane, David Romney, and Dustin Tingley. 2015. "Anti-Americanism and Anti-Interventionism in Arabic Twitter Discourses." *Perspectives on Politics* 13(1): 55–73.

Key, V.O. 1949. *Southern politics in state and nation*. New York: A. Knopf.

Kim, In Song, John Londregan, and Marc Ratkovic. 2014. Voting, speechmaking, and the dimensions of conflict in the US Senate. In *Annual Meeting of the Midwest Political Science Association*.

King, Gary, Christopher JL Murray, Joshua A Salomon, and Ajay Tandon. 2004. "Enhancing the validity and cross-cultural comparability of measurement in survey research." *American political science review* 98(01): 191–207.

Kingdon, John W. 1973. *Congressmen's voting decisions*. New York: Harper & Row.

Krippendorff, Klaus. 2013. *Content analysis: An introduction to its methodology*. Thousand Oaks, CA: Sage Publications.

Krosnick, Jon A. 1999. "Survey research." *Annual Review of Psychology* 50: 537–67.

Lauderdale, Benjamin, and Alexander Herzog. 2014. Measuring political positions from legislative debate texts on heterogeneous topics. Technical report Working Paper.

Laver, Michael, Kenneth Benoit, and John Garry. 2003. "Extracting policy positions from political texts using words as data." *American Political Science Review* 97(02): 311–331.

Lowe, Will, and Kenneth Benoit. 2013. "Validating estimates of latent traits from textual data using human judgment as a benchmark." *Political analysis* p. mpt002.

Lowe, Will, Ken Benoit, Slava Mihaylov, and M. Laver. 2011. "Scaling policy preferences from coded political texts." *Legislative Studies Quarterly* 36(1): 123–155.

Mikhaylov, Slava, Michael Laver, and Kenneth R Benoit. 2012. "Coder reliability and misclassification in the human coding of party manifestos." *Political Analysis* 20(1): 78–91.

Monroe, Burt L, Michael P Colaresi, and Kevin M Quinn. 2008. "Fightin'words: Lexical feature selection and evaluation for identifying the content of political conflict." *Political Analysis* 16(4): 372–403.

Neumayer, Eric. 2005. "Do international human rights treaties improve respect for human rights?" *Journal of conflict resolution* 49(6): 925–953.

Oishi, Shigehiro, Jungwon Hahn, Ulrich Schimmack, Phanikiran Radhakrishan, Vivian Dzokoto, and Stephen Ahadi. 2005. "The measurement of values across cultures: A pairwise comparison approach." *Journal of Research in Personality* 39(2): 299–305.

Owens, Ryan J, and Justin Wedeking. 2012. "Predicting drift on politically insulated institutions: A study of ideological drift on the United States Supreme Court." *The Journal of Politics* 74(02): 487–500.

Pang, Bo, and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics pp. 115–124.

Quinn, Alexander J, and Benjamin B Bederson. 2011. Human computation: a survey and taxonomy of a growing field. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM pp. 1403–1412.

Quinn, Kevin M, Burt L Monroe, Michael Colaresi, Michael H Crespin, and Dragomir R Radev. 2010. "How to analyze political attention with minimal assumptions and costs." *American Journal of Political Science* 54(1): 209–228.

Roberts, Margaret E, Brandon M Stewart, and Dustin Tingley. 2016. "Navigating the Local Modes of Big Data." *Computational Social Science* p. 51.

Roberts, Margaret E, Brandon M Stewart, Dustin Tingley, Christopher Lucas, Jetson Leder-Luis, Shana Kushner Gadarian, Bethany Albertson, and David G Rand. 2014. "Structural topic models for open-ended survey responses." *American Journal of Political Science* .

Sheng, Victor S, Foster Provost, and Panagiotis G Ipeirotis. 2008. Get another label? improving data quality and data mining using multiple, noisy labelers. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM pp. 614–622.

Slapin, Jonathan B, and Sven-Oliver Proksch. 2008. "A scaling model for estimating time-series party positions from texts." *American Journal of Political Science* 52(3): 705–722.

Snow, Rion, Brendan O'Connor, Daniel Jurafsky, and Andrew Y Ng. 2008. Cheap and fast—but is it good? Evaluating non-expert annotations for natural language tasks. In *Proceedings of the conference on empirical methods in natural language processing*. Association for Computational Linguistics pp. 254–263.

Socher, Richard, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*. Vol. 1631 Citeseer p. 1642.

Vempaty, Aditya, Lav R Varshney, and Pramod K Varshney. 2014. "Reliable crowdsourcing for multi-class labeling using coding theory." *IEEE Journal of Selected Topics in Signal Processing* 8(4): 667–679.

Von Ahn, Luis. 2009. Human computation. In *Design Automation Conference, 2009. DAC'09. 46th ACM/IEEE*. IEEE pp. 418–419.

Von Ahn, Luis, Benjamin Maurer, Colin McMillen, David Abraham, and Manuel Blum. 2008. "recaptcha: Human-based character recognition via web security measures." *Science* 321(5895): 1465–1468.

SI-1 ANALYSIS OF 50 MOVIE REVIEWS

We originally ran a similar movie review analysis as discussed in our first application using 50 reviews. We completed 60 comparisons in batches of 10. Several weeks later we collected another 20 comparisons. The findings are very similar to our application in the main text. Correlations peaked around 20 comparisons, and more comparisons increased precision but only mildly. However, with only 50 reviews every document was compared to every other document, or nearly so, making this analysis somewhat less instructive. As such we increased the number to 500 and repeated the exercise in the main text of our paper.

Nevertheless, we present the findings from this smaller study here. Figure SI-1 shows the correlations between stars at each successive iteration. Figure SI-2 shows the correlation between the first 20 and the last 20 comparisons estimated separately in the left panel, and the correlation between the first 20 and an additional round of 20 estimated separately.

Figure SI-1: Correlations after every ten comparisons

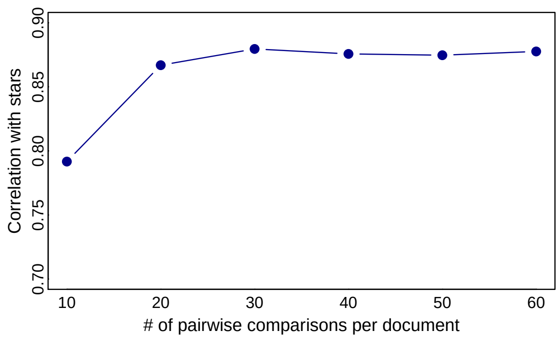
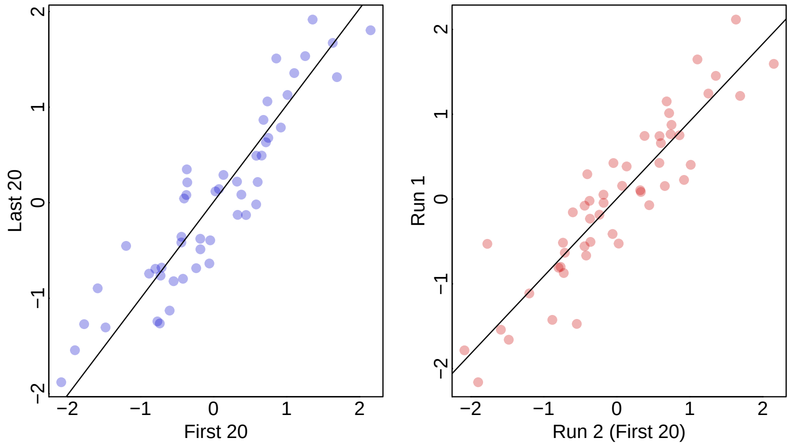


Figure SI-2: Reliability between first and last 20, and first 20 and separate experiment of 20



SI-2 BENCHMARKING AGAINST A MACHINE LEARNING METHOD

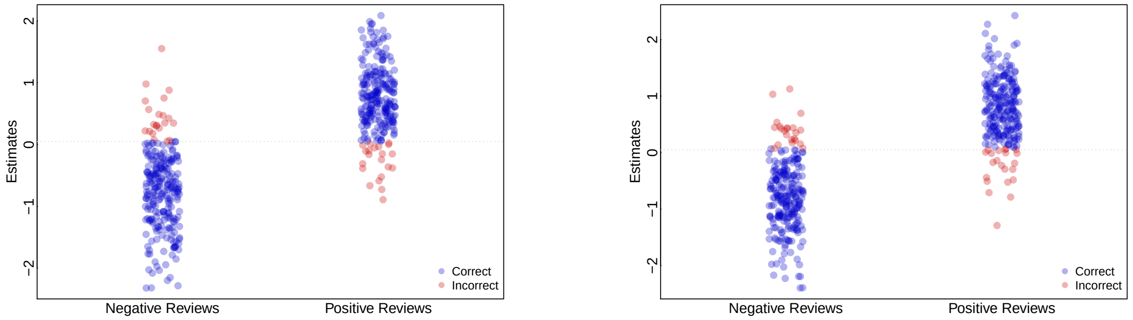
Pang and Lee (2005) analyze the polarity of 10,663 movie review snippets from Rotten Tomatoes. Half of the snippets (extracts from longer reviews) are taken from positive reviews, and half from negative reviews. Pang and Lee (2005) assume that snippets taken from positive reviews (e.g., 5-star reviews) are positive while the opposite is true for snippets taken from negative reviews.

This is a widely used corpus of texts in the computer science literature, and the success of various algorithms are often evaluated based on their success in classifying these statements. Socher et al. (2013), for instance, advertise the success of their supervised model by showing they are able to correctly classify 85.4% of the snippets as either positive or negative relative to the previous baseline of less than 80%. To illustrate the effectiveness of the `SentimentIt` approach we replicate their analysis with a random selection of 500 snippets, half of which are positive and half negative.

We did not choose to present the analysis in the main text because the exercise is aimed at categorization, and the short snippets do not fully leverage the utility of our approach. Further, we discovered that the training sets contained errors (e.g., positive snippets included in the negative grouping).[1]

We ran the experiment with and without a certification. We are able to classify 91.6% of the documents "correctly" without the qualification and 91.1% requiring a qualification. Figure SI-3 show the results of our experiment without and with qualifications. Most of the misclassified documents, upon further investigation, were found to be deceptive. They were largely either positive sentences taken from a negative review, or negative sentences taken from a positive review.

Figure SI-3: Identifying polarity in movie reviews without and with a qualification



Note: The left panel shows `SentimentIt` estimates without requiring a qualification and the right panel shows estimates requiring a qualification. The accuracies are nearly identical, with most misclassifications the same. This is likely due to the simplicity of the task.

[1] For instance, the statement, "The stunt work is top-notch; the dialogue and drama often food-spittingly funny," was included in the negative document set while the statement, "A brilliant gag at the expense of those who paid for it and those who pay to see it," was included in the positive document set.

SI-3 STAN CODE FOR STATISTICAL MODEL

We provide the Stan code for both the random utility model and the hierarchical random utility model in this section. This code was run in `R` using the `rstan` library. This is fully incorporated into our `R` package, `SentimentIt`. Stan utilizes Basic Euclidean Hamiltonian Monte Carlo sampling which involves three tuning parameters. Stan automatically determines these parameters, although allows for manual setting. We do not manually set these parameters, but allow Stan to set them automatically. We do, however, increase the maximum treedepth from the default to 50 for the hierarchical model. These options can be passed in our package.

Random utility model

```
data {
  int N; // number of comparisons
  int M; // number of documents
  int P; //Number of coders
  int y[N]; // outcome
  int g[N];    // id  map first item in comparison
  int h[N];    // id map of second itein comparison
  int j[N]; // id map for workers
}
parameters {
  real a[M];
  real<lower=0> b[P];
  real<lower=0> sigma;
}
model {
  sigma~normal(0,3);
  for(p in 1:P){
    b[p] ~ normal(0,1);
  }
  for(m in 1:M){
    a[m] ~ normal(0,sigma);
  }
  for(n in 1:N) {
    y[n] ~ bernoulli(inv_logit(b[j[n]]*(a[g[n]]-a[h[n]])));
  }
}
```

Hierarchical random utility model

```
data {
int N; // number of comparisons
int M; // number of paragraphs
int D; // number of documents (countries)
int P; //Number of coders
int y[N]; // outcome
int g[N];    // id  map first item in comparison
int h[N];    // id map of second item in comparison
int j[N]; // id map for workers
int k[M]; // id map for documents (countries) relating to documents
}
parameters {
```

```
real a[M]; // paragraphs
real t[D]; // documents (countries)
real<lower=0> b[P];
real<lower=0> sigmac[D];
}
model {
for(p in 1:P){
b[p] ~ normal(0,1);
}
for(d in 1:D){
t[d] ~ normal(0,1);
sigmac[d] ~ normal(0,.5);
}
for(m in 1:M){
a[m] ~ normal(t[k[m]],sigmac[k[m]]);
}
for(n in 1:N) {
y[n] ~ bernoulli(inv_logit(b[j[n]]*(a[g[n]]-a[h[n]])));
}
}
```

SI-4 ROBUSTNESS TO MODELING CHOICES

Table SI-1 shows the correlations between the `SentimentIt` estimates and the mean coder selection. Mean coder selection is calculated by averaging how often a document is chosen over the other documents in the comparisons. These correlations are very high, at or above 0.95. This demonstrates that our modeling choice is not driving the results or imposing restrictive assumptions and that our results are robust to the specific modeling and prior choices in the main text.

Table SI-1: Correlations between `SentimentIt` scores and mean coder selection

Application	Correlation
Movie reviews	0.96
Campaign ads	0.96
Immigration survey	0.97
Human rights reports	0.95

Note: The mean coder selection of the documents is highly correlated with the `SentimentIt` estimates. This suggests our modeling choice is not driving any result.

SI-5 EVALUATING WORKER QUALITY AND RELIABILITY

In this section, we argue that the Amazon Mechanical Turk Workers provide high-quality and reliable responses to the binary evaluations administered by the `SentimentIt` system. As has been reported in multiple previous studies, our evidence demonstrates that there is little reason to question the quality of the workers when appropriate steps are taken to correctly monitor workers and analyze the resulting responses.

SI-5.1. *Previous studies*

The availability of online workforces has only recently been utilized for social scientific research, with its primary focus being on inexpensive survey experimentation (e.g., Bohannon 2011). However the scientific application of online workforces is increasingly directed towards data analysis. Amazon's Mechanical Turk (AMT) service, in particular, has been leveraged to code data related to natural language processing; speech and vision; sentiment, polarity, and bias; information retrieval; and information extraction (Callison-Burch and Dredze 2010). In the realm of natural language processing, tasks such as affect recognition, word similarity, word sense disambiguation, etc., AMT-worker codings have a high degree of similarity with codings provided by expert coders (Snow et al. 2008). Because the online workforces are fast and inexpensive, deficiencies in the reliability of non-expert coders can be ameliorated with multiple labelings of the data (Ipeirotis et al. 2014). and through independent assessment of the observations (Vempaty, Varshney, and Varshney 2014).

For example, researchers of relevance assessment (determining if documents are relevant to a pre-specified request) have had significant success on the AMT platform. As with labeling, relevance assessments from AMT workers are reliable at levels comparable with gold-standard measures, particularly when coders are presented with binary choices (Alonso and Mizzaro 2012). Moreover, the reliability of coding relevance assessments improves when splitting the document into smaller tasks, allowing for shorter individual completion times as well as overall completion time due to parallelizing tasks (Alonso and Baeza-Yates 2011).

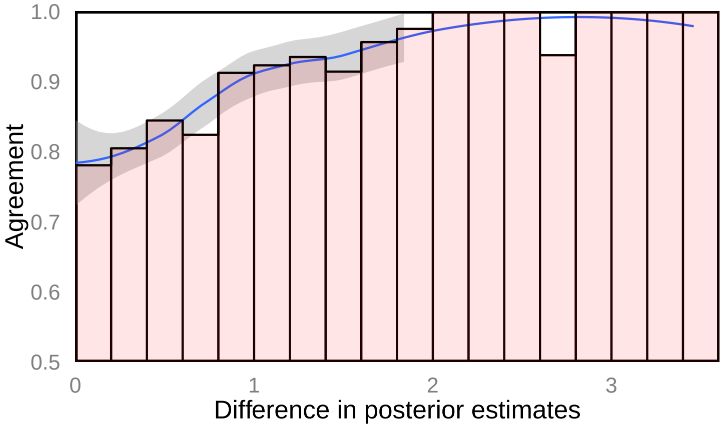
In the area of sentiment analysis, Hsueh, Melville, and Sindhwani (2009) compare AMT workers against expert coders to find that the aggregate accuracy of the online workforce approached the gold-standard set by the expert coders. This effect increased when removing the noisiest coders from analysis, suggesting that a screening process ought to improve accuracy.

SI-5.2. *Binary codings are reliable*

One way to check worker reliability is to assess the degree of agreement between repeated comparisons. To do this, we analyze the data shown in Appendix SI-1, where 50 movie reviews were evaluated in 60 different pairwise comparisons. This allows us to have a large number of instances where the exact same comparisons were made multiple times. In the application, 1049 HITs were duplicates, with 423 unique comparisons done more than once. (The same comparison was presented to coders from two to six times.) At the comparison level, 73.8% of duplicated comparisons had total agreement, meaning that all coders made the exact same decision. Thus, only 111 of the unique comparisons had some amount of disagreement, with 67 of those being 50-50 splits. In all, the average agreement on coding of these comparisons is 88.84%, meaning that in almost nine out of ten coding decisions of the same comparison were evaluated identically.

Of course, the fact that there is some disagreement is not surprising. Specifically, we would expect that documents that are very similar in terms of their level of positivity would be more difficult to distinguish leading to more disagreement. Figure SI-4 shows the proportion of agreement as a function of the absolute distance of their posterior estimates. In general, as the distance between the estimates increases, the proportion of agreement increases. Thus, when documents are very similar in their level of positivity, the average level of agreement is roughly 80%. However, once the documents become more distinguishable the agreement rate shoots up to over 90% or better for absolute distances of one or more.

Figure SI-4: Worker agreement on the difference of posterior estimates



Note: The plot shows the proportion of agreement of repeated comparisons on the difference of the posterior estimates. The blue line is a loess smoothed plot of the data points, and the red bars are binned average agreement with differences of 0.2. In general, as the difference in the posteriors increases, the proportion of agreement also increases.

SI-5.3. *Pairwise comparisons are transitive*

A further check to ensure workers are reliably completing the tasks is to check for transitivity in the pairwise comparison. This is also a useful check to ensure uni-dimensionality in the task. We analyze the application of 50 movie reviews described in Appendix SI-1 to increase the number of possible intransitive relationships since more complete triads appear in this dataset.

To assess the transitivity of the comparisons, we begin by generating a data frame of all available comparison triads. That is, we identified all cases where three documents had all been evaluated against each other (e.g., A-C, B-C, A-B). For each row, we then created columns, “codeA, codeB, and codeC”, that take on the following interpretations of their values:

$$\text{codeA} = 0 : A < B; \quad 1 : A > B$$
$$\text{codeB} = 0 : B < C; \quad 1 : B > C$$
$$\text{codeC} = 0 : C < A; \quad 1 : C > A$$

With this setup, it is clear that the only violations of transitivity occur when all three codes take on a value of 0 ($A < B < C < A$) or 1 ($A > B > C > A$). Therefore, to calculate the percent of cases that maintain transitivity, we simply calculate the percent of rows in the data frame of triads where the coded values do not sum to 0 or 3. Of 3,649 available triadic comparisons, 92.79% are transitive. We find these results both in line with uni-dimensionality and a competent workforce.

SI-5.4. *No evidence of systematic worker biases*

One potential problem for approach is there exist systematic biases among coders. Due to our procedure, idiosyncratic biases by subsets of coders are largely irrelevant. If one coder, for instance, tends to understand all ads as more negative than other coders, this is not relevant due to the pairwise comparison framework. So long as we are asking for only *relative* evaluations of ads, a general bias towards perceiving higher levels of negativity are irrelevant so long as it is applied equally to all documents. Somewhat more problematic is if a coder has a bias against subsets of documents. A liberal coder might, for instance, be more sympathetic to Democrats and view Republican ads as more negative in tone on average. However, in this case the statistical model in the main text will automatically down-weight her choices as they do not align with how other coders are evaluating the same documents.

However, a more serious problem is the possibility that, because these workers are not representative of the population, they may have biases in the aggregate. Imagine, as an example, that a large proportion fo the workers are biased against Republicans, something that is not impossible given the general liberal leanings of AMT workers (Berinsky, Huber, and Lenz 2012). In this hypothetical case, biases may persist both the pairwise comparisons framework and the statistical post-processing.

While we cannot entirely rule out this possibility in all instances, we have so far not found any patterns in our data supporting this claim. For instance, when we look at the most extreme estimates in the congressional ad application, there appears no relationship between the partisan affiliation of the advertisement and the direction or extremity of our estimates. The means of the Democratic ad estimates is 0.02 while the mean for Republicans is -0.01, both very close to zero and close to each other. Further, the correlation between the estimates and whether or not the ad is supporting a Democrat is 0.02 while the correlation of the estimates and the ad supporting a Republican candidate is -0.02. Of course, the parties may engage in attack ads with different intensities or frequencies, but we find no evidence that our estimates are at all related to the party of the advertisement as would be the case if the liberal biases of AMT workers were affecting evaluations.

SI-5.5. *Completion times*

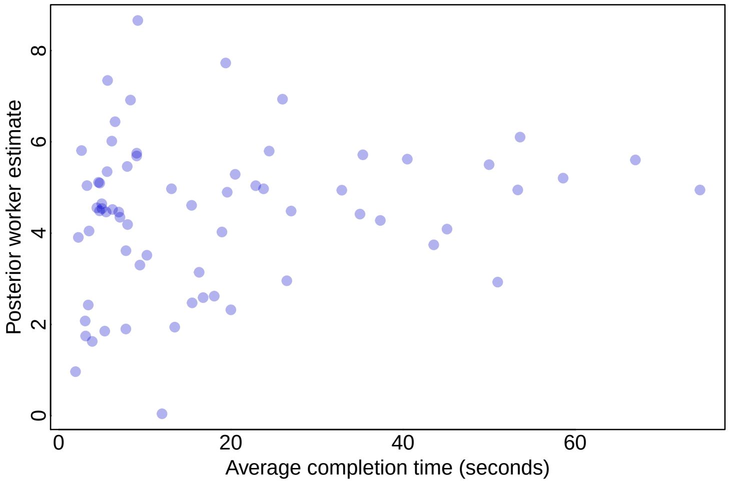
Despite the evidence that AMT workers are high quality workers, particularly when requiring training and a certification, it is possible that workers, once certified, are opportunistic and simply attempting to click through as many HITs as possible to accumulate the most money before (and if) we remove them from our jobs. To assess this possibility, we analyze the posterior worker estimates as a function of estimated HIT completion time. Figures SI-5 and SI-6 shows the posterior estimates of worker reliability plotted against the estimated average completion time for two of our applications (recall that lower estimates indicate lower-quality coders).[2] As can be seen, absolutely no pattern emerges, indicating that speed of completion may be more related to the abilities and pacing of the worker than the quality of their evaluations.[3] Likewise, Figure SI-7 shows the variance of the posterior estimates for the workers on their average completion time in the movie review application. No strong relationship seems present, suggesting further that completion time is not a significant determinant in worker quality or our uncertainty of the estimates. Thus, al-

[2] Plots for the remaining applications look essentially identical.

[3] Based on email conversations with workers, many begin to increase speed with practice as they become more accustomed to the task structure.

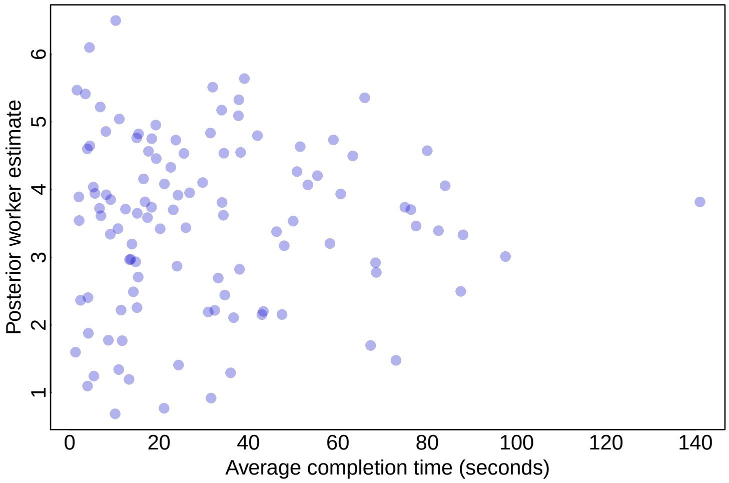
though we acknowledge that a small number of workers may act opportunistically if unmonitored, there is no evidence that a significant number of "bad" workers are simply clicking through HITs.

Figure SI-5: Posterior worker estimates on average completion time: Movie review application



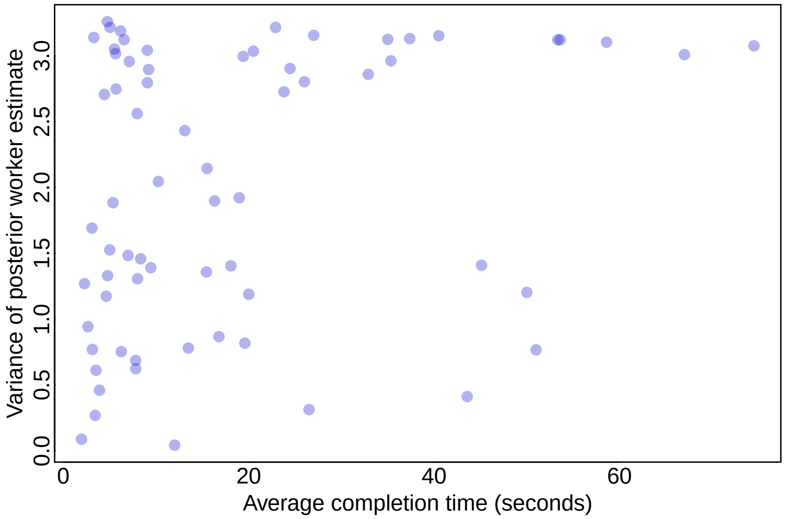
Note: There appears no relationship between average completion time and worker quality.

Figure SI-6: Posterior worker estimates on average completion time: Congressional advertisement application



Note: There appears no relationship between average completion time and worker quality.

Figure SI-7: Variance of posterior worker estimates on average completion time: Movie reviews application



Note: There appears no relationship between average completion time and the variance of the worker posterior estimates.

SI-6 TRAINING MODULE FOR HUMAN RIGHTS APPLICATION

Below is an example of a training module used in the `SentimentIt` platform. This is the training we utilized for the human rights application regarding torture. The module is meant to train and certify workers to answer the following question:

> Which of the two statements show more significant levels of torture? Torture is more significant if it there is evidence it is more frequent, more severe, unpunished, or systematic. Legal punishments and maltreatment of prisoners that is not used to intimidate or extract confessions are not considered to be torture.

The workers were presented with this question and two paragraphs from the human rights reports. Their task was to select the paragraph indicating greater degrees of torture. They were required to successfully complete this module in order to perform the task. If they failed to pass the qualification, they could not participate and could not retake the qualification.

SI-6.1. *Explanation of coding task*

If you finish this training module with a passing score, you will be qualified to complete HITs posted by the requester **SentimentIt** with the title **Compare Human Rights Report Extracts**.

This task involves comparing two text extracts from human rights reports on different countries detailing torture and prison conditions in that country. The texts are drawn from United States Department of State Country Reports on Human Rights from a particular year. Your job is to determine which extract indicates more **significant levels of torture**. What is meant by "more significant levels of torture" is explained below.

Each text extract is one complete paragraph with information on the state of torture, abuse, and prison conditions in a particular country. The country in question, as well as political groups, ethnic groups, and individuals may be named in the text extract, but please do not use your own knowledge to inform your decision in comparing the two texts. Instead, use only the information in the two texts displayed, and judge which text indicates more significant levels of torture by the following standards.

An extract indicates more significant levels of torture if:

- There is evidence of **more frequent** beatings, abuse, torture, and extrajudicial killings (killings that are not carried out in accordance with the legal procedures of the country); There is evidence of **more severe** instances of beatings, abuse, torture, and extrajudicial killings;
- There is evidence that when acts of torture are committed, they are **ignored, unpunished, or encouraged**, or that the use of torture is **routine, widespread, or systematic**;
- There is evidence that maltreatment or abuse in prisons is **specifically aimed at intimidating, penalizing, or obtaining a confession from detainees**;
- The **evidence given is well substantiated**, or better supported by reports from independent agencies (Nongovernment Organizations, The US State Department, etc.).

An extract **does not** indicate more significant levels of torture if:

- The punishment is **pursuant to the legal system** or standard legal practices of the country, even if some people might consider such practices as torture;
- **Prison conditions are poor or inadequate**, for example if there is overcrowding, inadequate food, or lengthy detention without trial;

- Allegations are specifically noted by the report to be **unsubstantiated or unlikely to be true**.

For each HIT, you will see text from **two** text extracts. Your task is to read both and select which of the two extracts indicates more significant levels of torture. That is, **based only on the two text extracts, which country do you think has more severe torture practices**?

It is important that you read each text extract carefully, and that you judge each by the standards listed above, and based only on the information in the text. **Do not** make your judgments on your own knowledge of the countries in question, on text extracts from previous HITs in this exercise, or on definitions of torture different to those listed above. Skimming or reading quickly will result in low-quality evaluations, and you may not be invited to participate in our future studies.

This training module has two parts. In Part 1, we will provide three practice HITs followed by instructions about how the text extract should be coded. In Part 2, we will give you six example HITs to complete. To receive the qualification for the Compare Human Rights Report Extracts task, you must complete five out of six of these example HITs correctly.

SI-6.2. *Example practice task*

This is your second practice Compare Human Rights Report Extracts HIT. Your answer will not be scored. Please read the two statements below and click on the button that corresponds with the statement which demonstrates more evidence of torture.

Which of the two statements show more significant levels of torture? Torture is more significant if it there is evidence it is more frequent, more severe, unpunished, or systematic. Legal punishments and maltreatment of prisoners that is not used to intimidate or extract confessions are not considered to be torture.

Statement A: According to defense attorneys and former prisoners, prison conditions ranged from Spartan to poor and, in some cases, did not meet minimum international standards. Credible sources reported that overcrowding continued to be a serious problem, with 40 to 50 prisoners typically confined to a single 194-square-foot cell and up to 140 prisoners held in a 323 square-foot-cell. A defense attorney reported that his client was imprisoned in a cell that contained 140 prisoners who were forced to sleep 3 to a cot. Defense attorneys reported that prisoners in the Ninth of April prison in Tunis were forced to share a single water and toilet facility and a single razor with their cellmates, creating serious sanitation problems.

Statement B: There are credible reports that torture occurred in prisons under the control of both the Taliban and the Northern Alliance. Local authorities maintain prisons in territories under their control and reportedly established torture cells in some of them. The Taliban operate prisons in Kandahar, Herat, Kabul, Jalalabad, Mazar-i-Sharif, Pul-i-Khumri, Shibarghan, Qala-e-Zaini, and Maimana. The Northern Alliance maintains prisons in Panjshir and Taloqan, and there also is a prison in the north at Faizabad, in Badakhshan province. According to Amnesty International, there have been reports that the Taliban forced prisoners to work on the construction of a new story on the Kandahar prison, and that some Taliban prisoners held by Masood were forced to labor in life-threatening conditions, such as digging trenches in mined areas.

After coders make a choice, the following text is shown. Relevant text in the statements are highlighted to make the decision criteria clear.

[Correct/incorrect]. Statement B shows more evidence of torture. While both statements describe harsh prison conditions, Statement B specifically mentions torture. Statement A describes severe overcrowding and maltreatment, but does not indicate that this is specifically designed to intimidate or extract confessions.

SI-6.3. *Example scored task*

Your answer to this example HIT will be scored. To receive the Compare Human Rights Report Extracts qualification, you must assess at least five of the six comparisons correctly. Please read the two statements below and click on the button that corresponds with the statement which demonstrates more evidence of torture.

Which of the two statements show more significant levels of torture? Torture is more significant if it there is evidence it is more frequent, more severe, unpunished, or systematic. Legal punishments and maltreatment of prisoners that is not used to intimidate or extract confessions are not considered to be torture.

Statement A: Methods of torture included electric shock, beatings (especially on the soles of the feet), suspension by the wrists or feet in contorted positions, burning, and near drownings. In other cases, victims are forced to remain in unnatural positions for extended periods, or have bags laced with insecticide, chili powder or gasoline placed over their heads. Detainees have reported broken bones and other serious injuries as a result of their mistreatment. There were no reports of rape in detention.

Statement B: Togo Security forces reportedly tortured a human rights monitor (see Section 4).

After coders make a choice, the following text is shown. Relevant text in the statements are highlighted to make the decision criteria clear.

[Correct/Incorrect]. Statement A contains more evidence of torture. While both statements indicate torture, Statement B reports an individual case of unknown severity, Statement A suggests widespread practice of very severe and methodical torture.

SI-7 SETTING UP CERTIFICATIONS

This section details the process of setting up a certification once the survey is completed in Qualtrics. The first step is to create a new qualification on Amazon. Then a new certification needs to be created through the `SentimentIt` API which will link to the Amazon qualification. Finally, the Qualtrics module needs to interact with the `SentimentIt` platform to grant the received certification, add workers who did not successfully complete the module to a different certification (the banned certification), and ensure that workers taking the survey have not received the banned certification.

SI-7.1. *Create a qualification on Amazon*

The first step is for the researcher to log in to Amazon Mechanical Turk as a requester. Once logged in, navigate to the "Manage" tab. Here there should be three links, the last of which is "Qualification Types." Click on this link and you will see a button "Create New Qualification Type." Create a new qualification by clicking this button. Two fields will appear, "Friendly Name" and "Description." Fill out the first with a meaningful name. This name will not be used by the `SentimentIt` platform but the workers will see this name under their received qualifications and when they see the posted HITs requiring this qualification. The second field should include a description of the certification and include a link to the qualification test. Again, workers will see this description when viewing the HITs. As an example, our congressional ad certification is titled "Senate Story Boards" and the description is:

> This task involves reading the text of two television advertisements aired during the 2008 U.S. Senate elections. Each advertisement consists of about one paragraph of text. Researchers will use your responses to better understand the "tone" of each political ad. To qualify for these HITs, you must complete a short training module located at: `https://wuslpolysci.co1.qualtrics.com/SE/?SID=SV_aWcYT6FeQbr8ZyB`

The link takes the worker directly to the Qualtrics survey.

A second qualification should also be created, one that is granted to workers who either fail the initial training or are banned by the researcher from participating in the tasks associated with the above qualification. For the same congressional certification we titled this second qualification "Banned Senate Story Boards" with the following description:

> This certification is granted if the Senate Story Board certification was not achieved, or the responses once certified were invalid. This is only applicable to tasks involving the Senate Story Board certification.

Once both qualifications are created, they are assigned identification codes by Amazon. These IDs are necessary for linking the MTurk qualification to the `SentimentIt` certifications.

SI-7.2. *Link the qualification to `SentimentIt`*

At this stage we can link the Amazon qualification to our certifications used in the `SentimentIt` platform with the unique ID retrieved from the previous step. On the `SentimentIt` API, navigate to Hit Setting/Certifications. Here there will be a button "New Certification." Once the button is clicked, a prompt for the name and Amazon ID will appear. The name should be something meaningful and short and contain no spaces. This is the name used by the `SentimentIt` platform and the R package. For the congressional ads, we used the name “congressads” and “bannedcongress” for the certification and banned list, respectively. The Amazon ID should be the exact ID given by Amazon for the certifications. Once these certifications are in the SentimentIt platform, they can be used in the HIT settings for comparisons.

SI-7.3. *Setting up the Qualtrics survey*

When the certifications are created, the Qualtrics survey can be altered to check if the worker is on the banned list as the training begins to disallow continuing, grant certifications to those successfully completing the training, and add workers who do not successfully complete the survey to the banned list. Following the prompt at the beginning of the survey requesting the worker ID, a page break should be inserted followed by a page with just a Next button. Our page reads: “Click the next button. It will be disabled if you have already taken the survey or have been banned.” On this page JavaScript code is embedded that checks if the worker is on the banned list (i.e., received the banned certification). The following code checks if the worker is banned from Senate Story Boards:

```
Qualtrics.SurveyEngine.addOnload(function()
{
    this.disableNextButton();
    var that = this;
    var worker_id = "${q://QID29/ChoiceTextEntryValue/1}";
    var get_url = "https://www.sentimentit.com/api/certifications/
        bannedcongress/turk_workers/"+worker_id+".json";
$j.ajax({
        type: 'GET',
        url: get_url,
        success: function (data) {
         if(data.allowed == true){
                alert('You have already taken this survey and cannot
                    continue.');
            }else{
                that.enableNextButton();
}
        },
        error: function (data) {
            that.enableNextButton();
     }
});
});
```

This code disables the next button and checks if the worker is on the banned list, here named “bannedcongress,” through a GET curl command. If the worker is not on the list, the next button is enabled. This name needs to be altered to match the banned certification of interest. The worker_id variable may need to be altered to match the specifics of the survey in question, but in our formatting this is simply the first text field in the survey asking for the MTurk worker ID. This code could easily be extended to check multiple certifications, such as a master banned list that tracks all workers the researcher would never want participating.

Once the training is completed, there should be a pass block and a fail block in the Qualtrics survey. In the failed block, we grant the worker the banned certification with the following JavaScript:

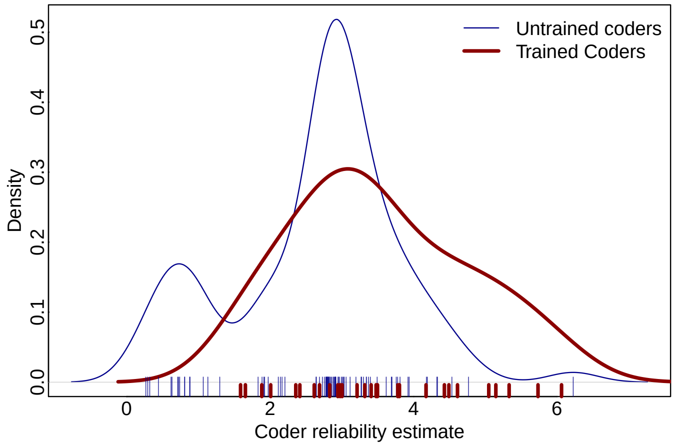
```
Qualtrics.SurveyEngine.addOnload(function()
{
    var worker_id = "${q://QID29/ChoiceTextEntryValue/1}";
    var input_data = "{\"certification\": \"bannedcongress\",
        \"workers\":[\""+worker_id+"\"]}";

    $j.ajax({
        type: ’POST’,
        url: "https://sentimentit.herokuapp.com/api/
            certifications/create.json",
        contentType: ’application/json’,
        //the following part is for when we have the username:password
            //security setup
        //beforeSend: function(xhr) {
        // xhr.setRequestHeader("Authorization", "Basic " +
            // btoa("username:password"));
        //},
        data: input_data,
        success: function (data) {
            console.log(data);
        },
        error: function (data) {
            console.log(data);
        },
        processData: false,
        //async: true
    });
});
```

Again, this code needs to be altered in order to match both the `worker_id` variable and the banned certification name. It grants the worker the certification through a `POST` curl command. In the passed block, the code is identical to this except the name of the certification should be the name of the certification required to complete the task (e.g., “congressads”).

SI-8 EVIDENCE ON THE ADVANTAGE OF TRAINING MODULES

In Appendix SI-2, we briefly discussed an experiment where we coded snippets from movie reviews using untrained and trained workers. Figure SI-3 shows that the classification rate was not noticeably affected by the training (possibly due to the simplicity of the task). However, athough requiring a qualification test did not improve our accuracy given the simplicity of the task, we found that the workers in the task requiring qualifications were of higher quality. (We did not ban any workers throughout the process.) Figure SI-8 shows the worker-level estimates. Clearly, requiring a simple qualification test disincentivised some low-quality workers from participating.

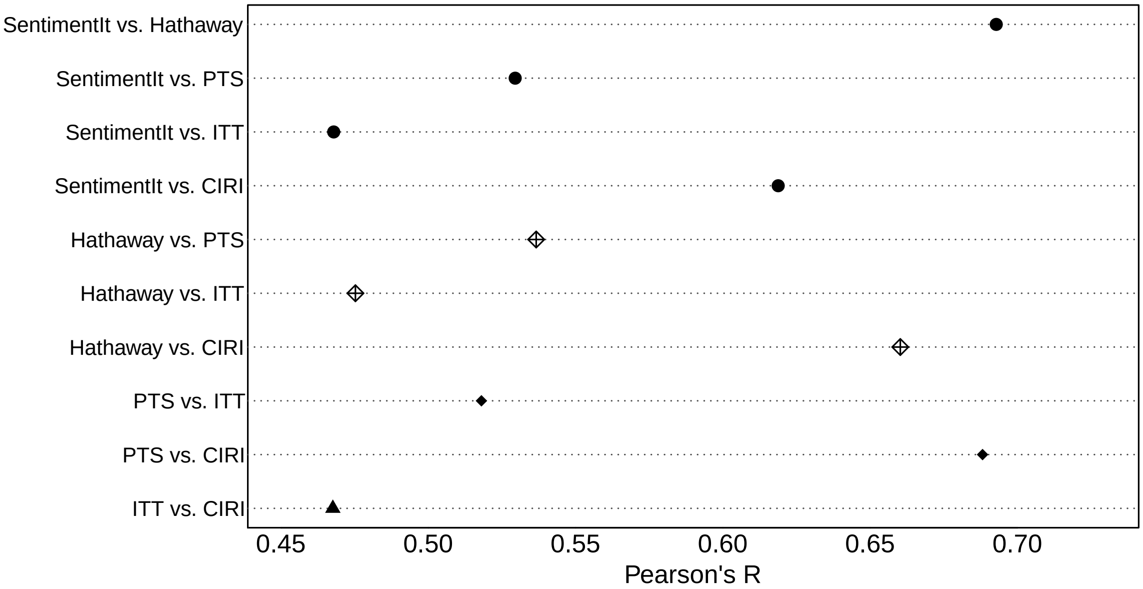
Figure SI-8: Worker-level estimates of untrained and trained workers



Note: Worker estimates in the experiment requiring training and a qualification shows that workers are performing more reliably than when no training is required.

SI-9 ALTERNATIVE MEASURE OF TORTURE

Figure SI-9: Correlations of various measures of torture



Note: CIRI correlations are the negative correlations because this measure is reverse coded. The highest correlation is between `SentimentIt` and Hathaway.

In the main text, we compare our measure with three other well-known measures of torture to assess the validity of these estimates as capturing the true degree of torture within the countries. First, we analyze the State Department variable of the Political Terror Scale (PTS) data on human rights violations (Gibney et al. 2015). This variable is constructed by a team of experienced coders, accounting for both the Human Rights Reports documents and other relelvant information. Mark Gibney and Reed Wood each code every country, and several other coders also read certain countries. When there is disagreement they discuss their decisions to reach a consensus. The second measure is the the Ill-Treatment and Torture (ITT) data (Conrad and Moore 2012). This measure is created from content analysis of Amnesty International's Annual Reports, press releases, and Action Reports. Multiple expert coders follow a very strict coding procedure. Finally, we use the torture variable from The CIRI Human Rights Dataset (Cingranelli, Richards, and Clay 2014). These data are coded by at least two expert coders, with any disagreements discussed with senior staff. Human Rights Reports and Amnesty International's Annual Reports are used for the coding. Figure SI-9 shows how all of these measures are correlated.

SI-10 `R` CODE USING OUR PACKAGE `SENTIMENTIT`

We provide as an example our first application to movie reviews. This data is also example data in the `R` package. We first show how interacting with online workforces can be done in stages to give an intuition behind the process and introduce the base functions used in our package. We then show how the entire process can be done using a wrapper function that will automate the entire process as a single command.

SI-10.1. *Basic functionality*

To begin, we first read in the text data, send it to the server, retrieve the document identification numbers, append these to the data, and write the data to a new file. The call is:

```
readText(email = ’researcher@school.edu’,
    password = ’uniquePassword’,
    read_documents_from = ’Reviews.csv’,
    write_documents_to = ’ReviewsWitdIds’,
    sep = ’,’, index = ’Review’, header = TRUE, quote = ’"’)
```

The `read_documents_from` argument is the file and path in which the text data is stored. This could optionally be a dataframe rather than a file path. Since this data is included as an exmple in the package, the following would load the data and perform the same operation:

```
data(reviews)
readText(email = ’researcher@school.edu’,
    password = ’uniquePassword’,
    read_documents_from = reviews,
    write_documents_to = "ReviewsWithIds",
    index = ’Review’)
```

The `write_documents_to` argument is the path and file name to store the data with the IDs appended. If set to the default, NULL, the function will return a data frame with the information rather than write to a file. The `what`, `sep`, and `quiet` arguments are used in the `scan()` function to read in the data when only text is provided, and all have defaults that would be standard for reading in datasets that contain only text. The `index` argument is the index indicating which column the text can be found, either the column name or column number. If an index is provided, `read.table()` is used instead, and the argument `sep` is used in this function. Not shown, the `which_source` argument is an indicator specifying from where the data came (e.g., “Rotten Tomatoes”). Optional arguments are all passed to the `scan()` function. In the example, we specify `quote=’"’` to indicate the quoting characters. Since the movie reviews contain quotation marks the specification is necessary. All of the functions that interact with the server require the researcher’s email and password used to register with `SentimentIt`.

Now that the data is on the server and we have the document IDs, we can create the batches of pairwise comparisons. We start with 10 comparisons per document, leading to 2,500 comparisons. We wish to send them out in batches of 1,000, so we need three batches. We run the following function:

```
batch_ids <- createBatches(email = ’researcher@school.edu’,
    password = 'uniquePassword',
    task_setting_id = 8,
    num_batches = 3)
```

This assigns batch_ids to the batch identification numbers returned from the server. The first argument following authentication, task_setting_id refers to the HIT setting we wish to use. These can be set on our server using a simple form to indicate what certification (if any) is required, how long workers have to complete the task once they have started, how much money they should be paid, and how long HITs should be posted for completion before expiration. In this case, the HIT setting requires a training certification, pays the workers $0.04 per HIT, allows the worker to take up to an hour answering the question, and leaves the HIT active on MTurk for 10 hours. This also dictates what the workers will see before selecting the HIT. In this case, they see:

> You will be asked to read some text from two movie reviews and decide which seems like a more positive review. To be qualified to complete this HIT, you must complete the training module at https://wuslpolysci.co1.qualtrics.com/SE/?SID=SV_b1LMfxcmlVOtOOF

The URL is where our training certification is located. If the workers follow the link and successfully complete the training, their worker ID will be automatically added to the list of approved workers and complete the associated HITs. If they fail the certification, they are banned from retaking the training or answering HITs associated with this setting.

Now that the batch settings are specified, we can create 10 random pairwise comparisons. We first need to retrieve the document IDs created earlier, written to the file specified, then create the comparisons using these IDs. The following code will set up the comparisons:

```
docInfo <- read.table("ReviewsWithIds", header=TRUE)
makeCompsSep(email = 'researcher@school.edu',
    password = 'uniquePassword',
    ids = docInfo[,'ids'],
    number_per = 10,
    batch_id = batch_ids,
    question = 'Below is text taken from two movie reviews.
            Please choose the text that you think comes
            from the most positive review',
    pairwise_path = 'Comparisons/first10.Rdata')
```

The first argument, ids, indicates the numerical IDs for the documents. The argument number_per is the number of comparisons desired (in this case 10). The batch_id argument indicates the batch IDs to be used for the HITs. The question argument specifies the question the worker will see once the worker selects the HIT. There is an argument per_batch indicating the number of comparisons per batch desired, defaulted to 1,000. If the number of comparisons is not a multiple of this number, the final batch will have fewer comparisons. We have 2,500 comparisons, so the final batch only has 500 comparisons. The number of comparisons per batch could have been automatically determined, but forcing this number to be provided ensures no mistakes or made, such as providing too few batches. The pairwise_path argument is used to save the comparisons that were created to a specified path and file name where the comparisons should be stored.

The function returns the batch IDs as returned by the server (which we already have stored). This serves as an assurance that the function has been correctly called.

Now that the comparisons are set up and on the server, we can post them as HITs to AMT. If we wish to send our first batch, we run:

```
createTasks(email = 'researcher@school.edu',
     password = 'uniquePassword',
     batch_id = batch_ids[1])
```

The argument `batch_id` is the identification number for the batch that we want to send, which we retrieved from the call to `createBatches()`.

At this point, we want to occasionally check the status of a batch. That is, how many of the comparisons have been completed. To do this we run the function:

```
batchStatus(email = 'researcher@school.edu',
     password = 'uniquePassword',
     batch_id = batch_ids[1])
```

The only argument to this function, other than authentication, is `batch_id`, the ID of the batch you wish to check. This could be a vector of batch IDs. This returns a dataframe with the batch ID and the number of comparisons total, submitted, completed, and expired.

Once the batch is completed (or near completed), we can check the workers to find any that are deviant. First, we need to read in the data from the server. We accomplish this by running:

```
output <- readInData(email = 'researcher@school.edu',
     password = 'uniquePassword',
     batch_id = batch_ids[1])
```

The argument to this function is `batch_id`, which could be a scalar or a vector of batch ID numbers. The returned output is a data frame with the following columns: batch_id, comparison_id, document_id, result, hit_id, worker_id, and completed_at. The data is organized by document-comparison, and the result is an indicator if the document was chosen over the other document in the given comparison. (There are, therefore, two rows for every comparison conveniently grouped by comparison so every odd row is immediately followed by an entry for the other document in the same comparison.) The worker_id is the AMT identification number of the worker, important for keeping track of the performance of workers to determine deviant (or highly reliable) workers. Finally, completed_at is a time stamp for the HIT completion time. This can be used to determine how quickly workers are completed tasks. If a worker is finishing HITs in a very fast amount of time this may suggest the worker is simply clicking through as fast as possible. In our experience only a very small proportion of workers do this, and it is quite obvious if workers are simply providing insincere evaluations.

We now need to fit the Stan model to estimate worker reliability. For the non-hierarchical model, which we used in this example, we run:

```
fit <- fitStan(data = output)
```

This function optionally has the following arguments with defaults:

```
fitStan(email = NULL, password = NULL,
    data, chains = 3, iter = 2500, seed = 1234, n.cores = 3)
```

The first two arguments are only necessary if the data provided are batch numbers rather than actual data. The following argument is the output from `readInData`, but can alternatively be a (vector of) batch ID number(s), allowing the researcher to skip the earlier step. The latter arguments are all used in the call to Stan through `rstan`. The first, `chains`, is the number of chains to run in the sampling process. The second, `iter`, is the number of iterations to run in the sampler. The default, 2500, is a conservative number that should ensure convergence even in larger datasets. However, the researcher may want to increase this number if computing time is not an issue so convergence does not need to be checked. The argument `seed` is the random seed used in the model fitting, allowing reproducibility. Finally, `n.cores` is the number of cores to run in parallel. If the researcher wants this entire process to be running in the background, changing the number of cores to use should be considered. If the process is running on a four core machine, for example, these defaults would only leave one core free for other processes.

There is a related function that fits the hierarchical Stan model. It takes the following arguments:

```
fitStanHier(email = NULL, password = NULL,
    data, hierarchy_data, hierarchy_var,
    chains=3, iter=2500, seed=1234)
```

The arguments are the same as `fitStan()`, but with two additional arguments, `hierarchy_data` and `hierarchy_var`. In order to fit the hierarchical model, the data used to set up the documents and comparisons needs to be provided in order to map the document IDs to their respective higher-level grouping. In the case of the human rights reports, this would be data that consisted of the paragraphs, the document IDs for the paragraphs, and a variable for the country of the reports from which the paragraphs came. The column name or index number for the country would be the argument for `hierarchy_var`, while the data itself would be `hierarchy_data`.

Once the model is fitted, we can check for outlier workers. To do so, we run:

```
ban_workers <- checkWorkers(stan_fit = fit, data = output)
```

The first argument is the `rstan` object obtained from the previous step, and the data is the output from the batch(es). The function has optional arguments with defaults. The full list of arguments is:

```
checkWorkers(stan_fit, data, cut_point=1, cut_proportion=0.9,
                n.questions=50, plot_hist=FALSE,
                hist_path=NULL)
```

The argument `cut_point` is the estimate cut-off for a worker to be considered an outlier. If the proportion of posterior draws falling below the cut-off point is greater than `cut_proportion`, and the worker has answered at least `n.questions`, the worker is considered an outlier. The function returns a character vector of the MTurk IDs of workers estimated as outliers. The argument `plot_hist` is an indicator to plot the histogram of workers with a rug plot so the researcher can visually inspect the distribution. The argument `hist_path` is an argument of the file name where the plot will be saved. If left as the default, NULL, the plot will only appear through the `R` session and will not be saved.

We want to revoke the certification for these workers and add them to the list of banned workers for this task. We keep track of banned workers by granting them a different certification that indicates their certifications have been revoked. This is also how we keep track of workers that fail the qualification. First, to revoke the certification, we run:

```
revokeCert(email = 'researcher@school.edu',
    password = 'uniquePassword',
    cert = 'snippets', workers = ban_workers)
```

The `cert` argument is the name of the certification as used on the server, which in this case is titled snippets for movie reviews. The next argument is a vector of worker IDs obtained from the previous step. We now add them to banned list:

```
createCert(email = 'researcher@school.edu',
    password = 'uniquePassword',
    cert= 'bannedmovie_reviews', workers = ban_workers)
```

This will grant the workers the certification bannedmovie_reviews which indicates they can no longer participate in HITs requiring the snippets qualification.

We can now proceed with posting the other batches, checking workers whenever we choose. Once all the batches are completed, we find it common that a few HITs remain incomplete with a few HITs per batch overlooked. We can then run:

```
repostExpired(email = 'researcher@school.edu',
    password = 'uniquePassword',
    batch_id = batch_ids)
```

This will repost all of the expired HITs from the vector of batch IDs. Finally, we can run `readInData()` and `fitStan()` to retrieve final estimates of all the data.

SI-10.2. *Higher-level functions*

The functionality discussed to this point is the lowest level of functionality, where the researcher has the most control. However, we also provide "wrapper" functions that make the process easier. The most comprehensive is the `sentimentIt` function, which automates every step of the process. Complete documentation of the software is beyond the scope of this paper, but below we provide an example of a call to this function.

```
data(reviews)
movies <- sentimentIt(email = 'researcher@school.edu',
    password = 'uniquePassword',
    read_documents_from = reviews,
    write_documents_to = 'ReviewsWithIds',
    index = 'Review', task_setting_id = 8,
    number_per = 10,
    question = 'Below is text taken from
```

```
          two movie reviews. Please
          choose the text that you
          think comes from the most
          positive review',
      pairwise_path = 'Comparisons.Rdata',
      certone = 'snippets', certtwo = 'bannedmovie_reviews',
      timed = FALSE, check_workers_at = c(1,2),
      rest_time = 60, rate = 1/3, threshold = 5,
      return_stan = TRUE, return_data = TRUE)
```

This command will perform the following function:

- All documents in the `reviews` dataframe will be read in and passed to our servers.
- HIT settings for the task will be assigned (question wording, compensation, duration, etc.) and the `snippet` certification will be required.
- Ten random pairwise comparisons per document will be created.
- All unique identifiers for documents will be stored at `ReviewsWithIds`
- Comparisons will be posted to AMT in batches of 1,000.
- New batches will be posted once the current batch is completed up until the `threshold` of five incomplete comparisons. Completion status will be checked every `rate=1/3` of an hour.
- A Stan model with three chains and 2,500 iterations will be fit when the workers are checked and when the final data is analyzed.
- Workers will be evaluated after the first 1,000 and second 1,000 comparisons are complete, and workers with 0.9 of the posterior draws falling below the default cut point of $b_k = 1$, will be banned from completing future tasks.
- After posting comparisons, the function will wait `rest_time = 60` seconds before posting HITs to Mechanical Turk to ensure all of the comparisons are ready to be posted.
- All incomplete tasks will be re-posted.
- After all tasks are completed, the data and Stan estimates of all model parameters will be returned.

There are several advantages to this higher-level approach. First and foremost, this functionality makes the process of interacting with online workers simple and efficient from the perspective of a researcher. Once the qualifications and HIT settings have been created, a single command can supervise the collection of worker evaluations and the creation of a meaningful measure – even if this process requires several days. However, a further advantage of this approach is that it makes the process of turning text into data highly replicable. Researchers wishing to evaluate the reliability of any measure can simply re-run the task using the original call above to create an exact replication of the original measure. Thus, the `SentimentIt` platform has the potential to bring about a higher degree of transparency to the task of turning natural language into meaningful data.

SI-11 UNCERTAINTY IN ESTIMATES

One concern readers may have is that documents scores in the middle of the spectrum may simply be more difficult to code rather than being located in the middle of the distribution. To examine this possibility, we examined the measures of uncertainty (posterior standard deviation) associated with each document in our datasets. In short, we find no evidence to support this contention. Rather, as is typical for random utility models (and other models closely related to item-response theoretic models) we find that uncertainty estimates are actually highest for the most extreme documents.[4]

As an example, figure SI-10 shows our estimates of the movie reviews, grouped by the star rating provided by the reviewer, with 95% posterior density bars. Note that the estimates at the extreme have larger density intervals, with longer tails towards the extremes. There is overlap in the intervals between star ratings, which we discuss in the text, but overall the majority of estimates do not have overlapping intervals with estimates as proximate as a two star difference. Also of note is the intervals do not seem to vary considerably between estimates except as a function of extremity. This again suggests that there are no discernible patterns in our posteriors that are not imposed by the modeling choices, coder unreliability, and the like.

Figures SI-11 and SI-12 show the degree of uncertainty in our posterior estimates plotted on the document estimates for the movie review and congressional advertisement applications respectively. There is a very clear pattern that as the estimates become more extreme in either direction, the uncertainty of those estimates increases as is typical in measurement models in this family.

Figure SI-13 show the standard deviations of the paragraph-level estimates for the human rights applications on the estimates. Again, we see that more extreme estimates have larger levels of uncertainty. Figure SI-14 shows the same relationship for the document-level estimates. Here there is no clear pattern in the highest density region. However, there is a clear decrease in uncertainty as the document estimates become much lower. This is sensible, because the documents estimated at the low extreme are short, containing only a few paragraphs that convey only positive information. The estimate of the document with the lowest uncertainty is, however, Cuba. In this document, all paragraphs are estimated very close to the document estimate, because each paragraph is an account of some torturous act. The document with the greatest amount of uncertainty is Nepal, with a document-level estimate of 0.36 with a standard deviation of 1.18. The paragraphs for this document have estimates ranging from -2.06 to 1.87, with some clearly positive statements such as stating that the government allows human rights visits and some clearly negative statements regarding widespread, brutal torture methods used to extract confessions.

[4] Speaking loosely, the posterior variance is most significantly reduced by comparing documents with others that are nearby in the latent space. By their nature, extreme documents are less likely to be compared with those nearby in the latent space leading to higher levels of posterior uncertainty.

Figure SI-10: Movie review stars on estimates with 95% posterior density

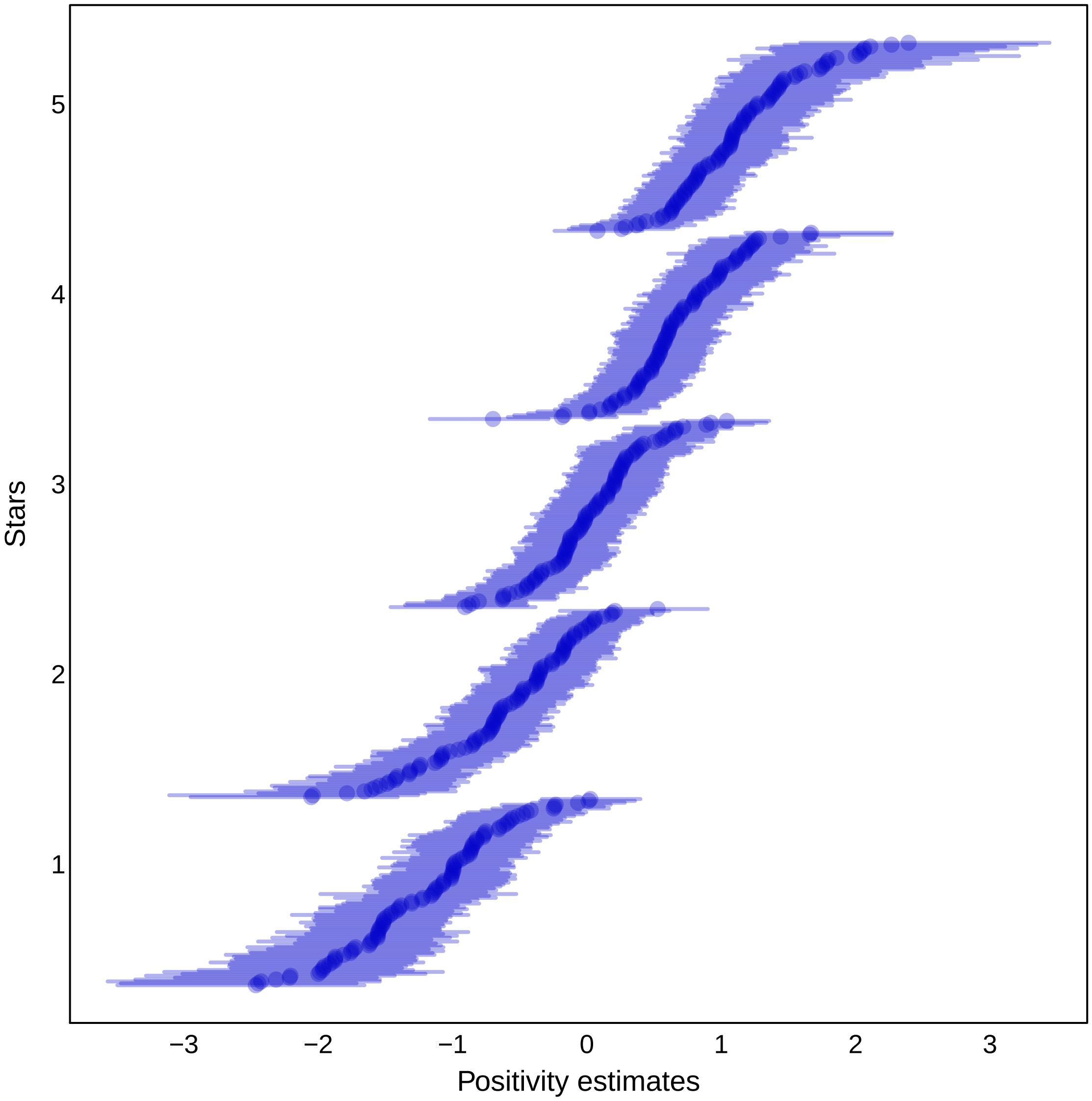


Figure SI-11: Movie review uncertainty: Standard deviations on positivity estimates

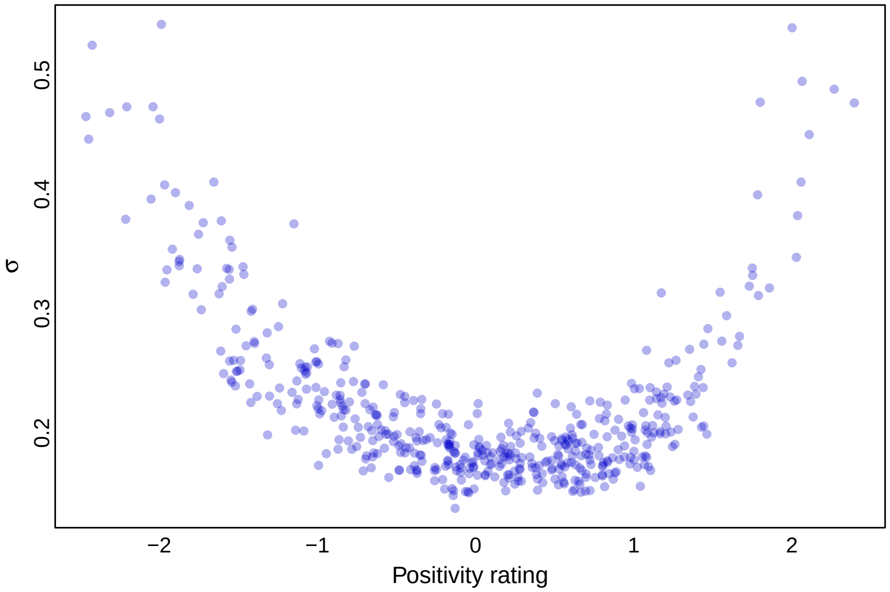


Figure SI-12: Congressional advertisements uncertainty: Standard deviations on negativity estimates

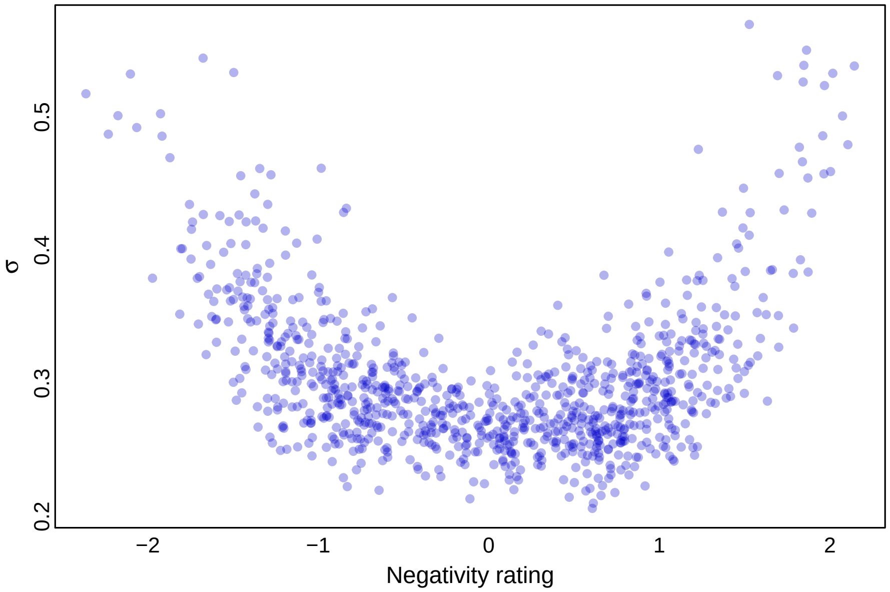


Figure SI-13: Human rights paragraph uncertainty: Standard deviations on torture estimates

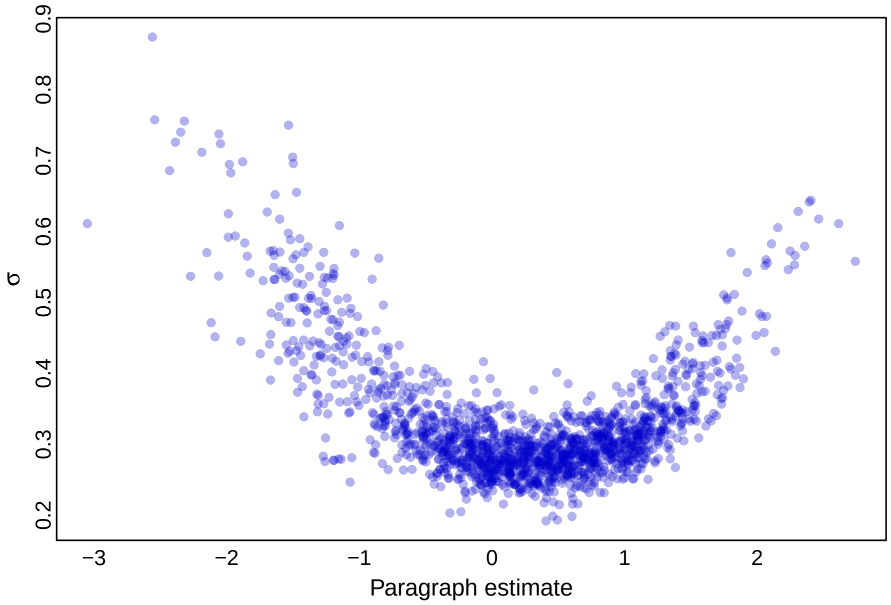
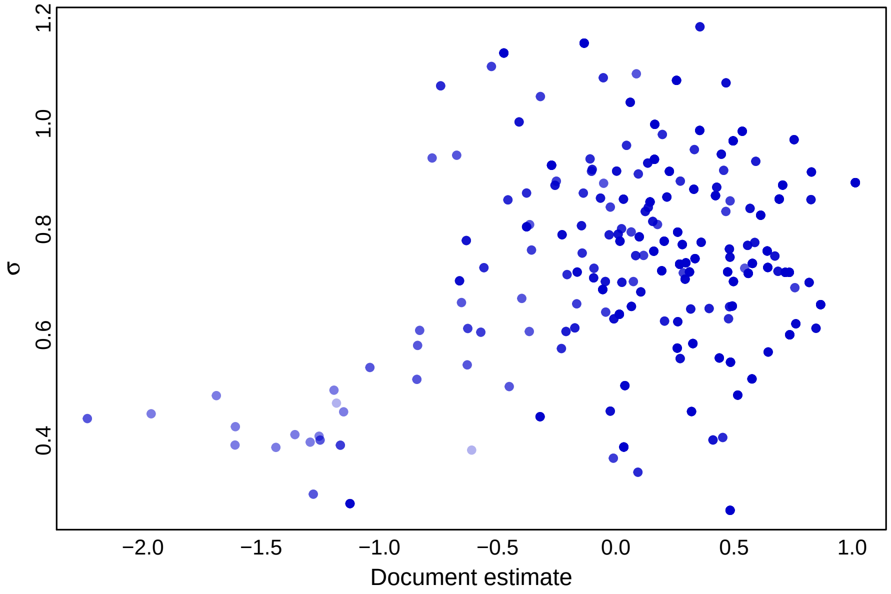


Figure SI-14: Human rights document uncertainty: Standard deviations on torture estimates


σ
Document estimate