A pairwise comparison framework for fast, flexible, and reliable human coding of political texts

Supplementary Information

David Carlson
Washington University in St. Louis

Jacob M. Montgomery
Washington University in St. Louis

In our letter, we provide an overview of the `SentimentIt` platform and examples of how it can be used. Due to space constraints, these are necessarily brief and cursory with regards to some details. In this Supplemental Information Appendix we provide additional details about our method, additional applications, and extensive discussions of practical concerns for researchers wishing to implement our approach on their own. The topics covered in this appendix are:

SI-1 ANALYSIS OF MOVIE REVIEWS

In our initial applications of the `SentimentIt` platform we focused on analyzing user-contributed movie reviews from the Rotten Tomatoes website along with their associated star rating. Rotten Tomatoes allows users to rate movies on a five-star scale. We selected random movie reviews from the website and then use our 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 correlated with the star ratings. However, we also wish to show that where disagreement exists, our measure provides a superior numerical summary of the texts' tone.

SI-1.1. *Study of 50 movie reviews*

First, we gathered 50 reviews. We completed 60 comparisons in batches of 10. Several weeks later we collected another 20 comparisons for a total of 80 comparisons for each review. Figure SI-1 shows the correlations between stars at each successive iteration. Correlations peaked around 20 comparisons at about 0.86, and more comparisons increased precision but only mildly.

Figure SI-1: Correlations after every ten comparisons

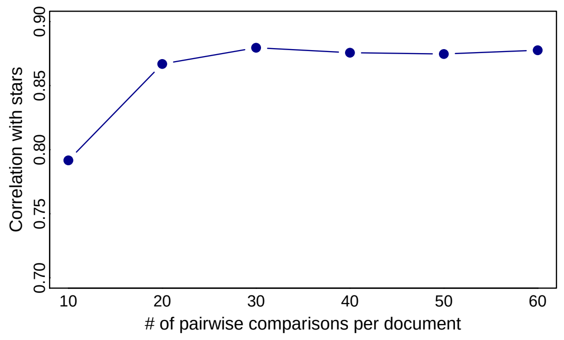
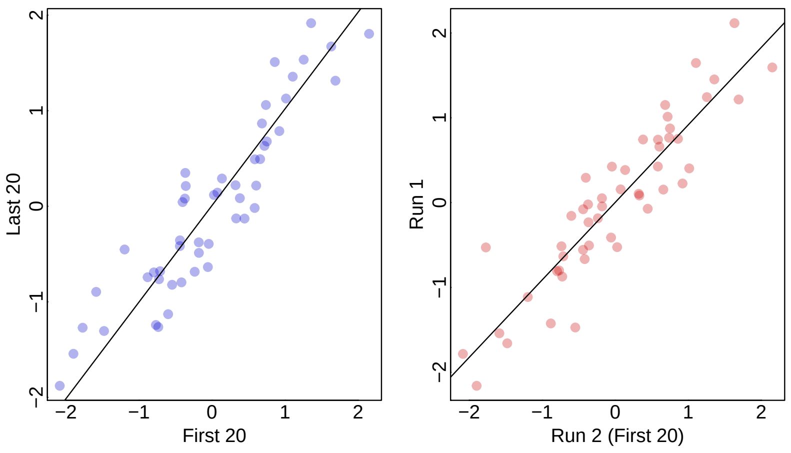


Figure SI-2 shows the correlation between the first 20 and the last 20 comparisons estimated separately in the left panel, and the correlation between the first 20 and an additional round of 20 estimated separately. As in the main text, these correlations are very high ($r > 0.90$).

SI-1.2. *Study of 500 movie reviews*

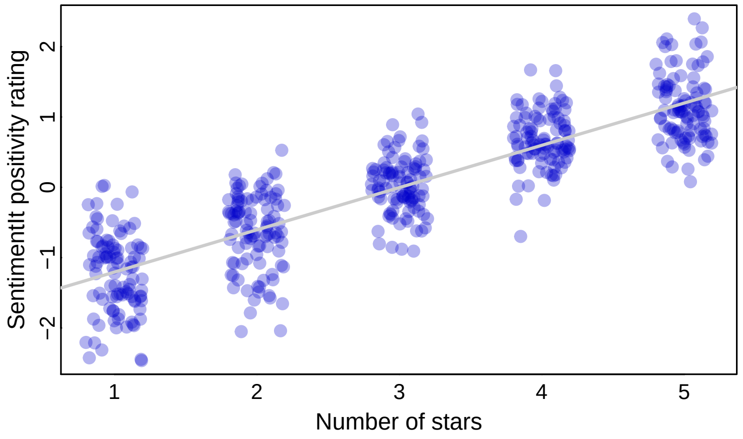
Next, we selected 500 user-contributed movie reviews from the Rotten Tomatoes website, choosing 100 reviews from each star category. We created 40 random pairwise comparisons per document, resulting in 10,000 comparisons. We required participating workers to complete a qualification, and paid them \$0.04 per task. We sent tasks to AMT in batches of 1,000 or 500 and

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



analyzed the results between each batch to identify low-quality workers. Throughout the experiment we banned only two workers out of 126.

Figure SI-3: `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.

Figure SI-3 shows our estimates plotted against the number of stars assigned by the author. The figure shows a very clear trend: as the stars increase so do our estimates (Pearson's $r = 0.87$). Further, in Table SI-1 we show 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 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 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.

SI-1.3. *Open-ended survey responses*

In our next application, we analyze a corpus of open-ended survey responses about immigration collected by Gadarian and Albertson (2014). Gadarian and Albertson (2014) used two trained research assistants to code each statement as indicating no (0), some (1), or extreme (2) levels of fear, anxiety, or worry towards immigrants or immigration. Coders were given specific instructions to distinguish between statements that indicated anger versus those that indicated fear, anxiety, or worry. They averaged the two coders' evaluations to generate a 2-point scale for each response. We analyze the survey responses with 40 pairwise comparisons per document in two separate experiments of 20 comparisons. We required a certification and paid \$0.04 per task. Each experiment was completed in about three hours. These experiments were approximately one week apart. In total, 20 workers participated and we banned two.

Table SI-2 shows five examples of such responses from the 342 collected.[1] Workers were instructed to choose which statement indicated the greater degree of fear, anxiety, or worry towards immigrants. The full prompt was:

> Which of these two statements expresses more fear, anxiety, or worry about the negative impact of immigrants or immigration on America? Remember, we are not interested in whether the writer dislikes immigrants, wants them to go home, resents them, or blames them. We are only interested in whether the writer is expressing fear, anxiety, or worry.

This prompt, as well as the required training, were designed in consultation with the original authors to replicate their coding scheme. That is, we specifically defined different emotional states to our coders in a manner similar to Gadarian and Albertson (2014). Thus, although our focus here is on the "fearfulness" dimension, special attention was given to helping coders distinguish this from related emotional states (anger) or general negative affect towards immigrants.

Figure SI-4 compares the `SentimentIt` estimates relative to the mean expert-coder rating. Figure SI-5 compares the correlations between `SentimentIt`, each expert coder, and the mean of the expert codes. The figures show that as the expert coding increases, the `SentimentIt` measure does as well. The `SentimentIt` scores are correlated with the mean expert rating at $r = 0.78$, a modestly strong correlation. Indeed, the `SentimentIt` measures correlate more highly with each expert coder than the individual coder scores correlate with each other.

[1]All capitalization was removed from these texts.

Table SI-1: Reviews with significant disagreement between `SentimentIt` scores and star ratings

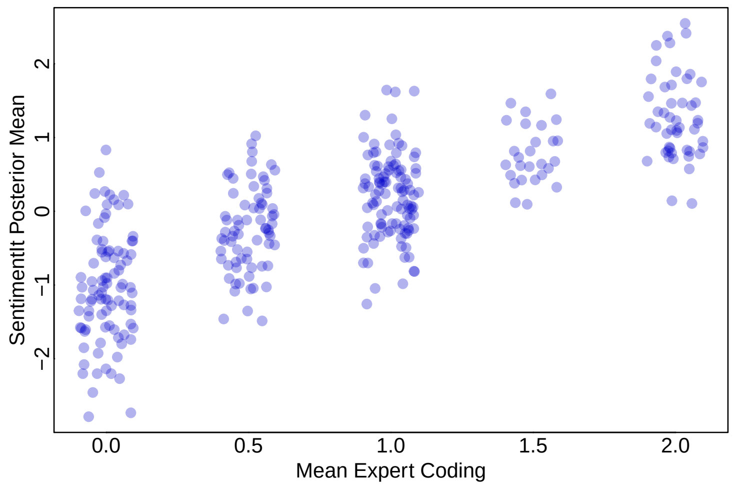
Review text	Positivity score	Stars
"I really enjoyed the original with Brooke Shields and yes it is on my movies that are very hard to find list, however deemed this sequel just ok. Somethings they should just leave as they are."	0.03	1
"Eddie Murphy. What happened to your taste in movie roles? This 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: Where `SentimentIt` uncovers meaningful differences in the tone of reviews ostensibly assigned the same star ratings by reviewers.

Table SI-2: Example open-ended survey responses regarding attitudes towards immigration

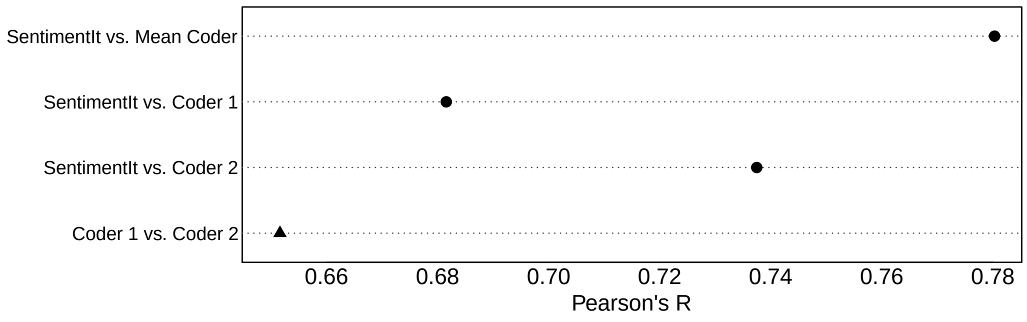
Survey responses
"terror bombings killing us robbing america"
"taking jobs from americans"
"illegals; social security; benefits"
"i worry about the republican party doing something very stupid. this country was built on immigration, to deny anyone access to citizenship is unconstitutional. what happened to give me your poor, sick, and tired?"
"the mexican border illegal aliens from other countries people getting a chance to have a decent life people like our country better than theirs"

Figure SI-4: Fearfulness estimates from `SentimentIt` and STM on expert coding



Note: The estimates from `SentimentIt` increase as do the human codes.

Figure SI-5: Correlations between different sources of fearfulness estimates



One advantage of the `SentimentIt` approach is that it provides scores on a continuous metric and can reveal important variation within the broad categories identified by the expert coders. Table SI-3 shows statements that were unanimously categorized by the expert coders (scored 0, 1, or 2). The statements listed are those which `SentimentIt` identified as being the most fearful or least fearful within each category. These examples illustrate that there is significant variation within categories (as identified by the expert coders) that is successfully identified by the `SentimentIt` approach. For instance, the system correctly identifies that a statement where a respondent indicates concerns that immigrants being "artificially low wage earner takes away jobs that could go to Americans" exhibits more fear than a statement that mentions less specific concerns about "all the illegals coming in to the country." Likewise, it correctly identifies statements like, "i am fine with immigration," as indicating far less anxiety than those that express concern about illegal immigrants who "think they have the right to drive" along with a preference for legal immigrants who "honor our country and pledge allegiance to our flag." In general, a close reading of the texts and the associated scores show that the `SentimentIt` system is not perfect, but it assigns scores to texts that broadly reflect the level of fear indicated in the text—in many cases more so than the scores assigned by the expert coders.

Finally, we turn to the question of reliability. Note that the two expert coders provide estimates that are correlated at only 0.65. This is partly a function of differences in how coders relate specific texts to given categories using an absolute scale—an extremely common feature of such coding schemes. Table SI-4 shows that the coders fully agreed on the categorization on only 67% of statements (the cells on the diagonal). Moreover, disagreements were not symmetric, with Coder 2 systematically placing more statements in more "fearful" categories relative to Coder 1. In comparison, the `SentimentIt` procedure's reliance on the pairwise comparison framework significantly reduces the influence of these kinds of coder-specific effects. Approximately one week 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. The correlation between these two `SentimentIt` analyses is a very impressive $r = 0.86$.

Table SI-3: Variation within categories uncovered by `SentimentIt`

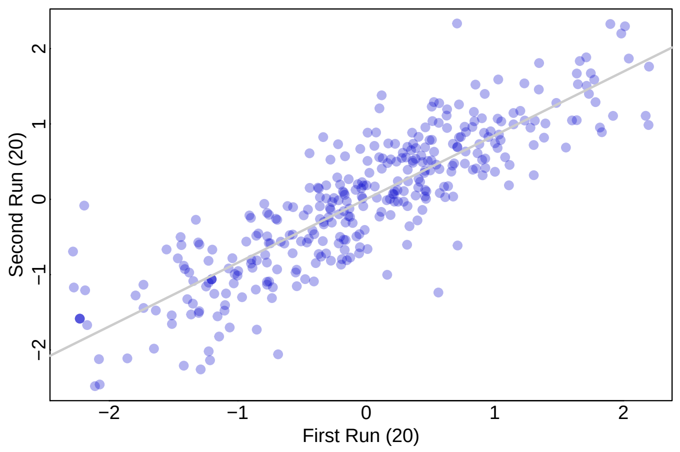
Survey response	Expert coders	`SentimentIt`
when i think of immigration i think of people who enter this country legally, who go through the proper immigration process, no matter how long it takes. i think of people who are willing to learn the english language, make an honest living, honor our country and pledge allegiance to our flag. those who come to america by any other means, who sneak in here and file false paperwork, who think they have the right to drive and have a license, who manage to obtain false ssn's don't deserve to be here, and our borders need to be much, much more secure.	None (0)	0.83
i am fine with immigration.	None (0)	-2.79
illegal immigrants coming into our country.illegal immigrants being a artificially low wage earner which takes away jobs that could go to americans.illegal immigrants working here but not paying their fair share of taxes since it is in large part an underground "cash based" society	Some (1)	1.64
all the illegals coming in to the country.	Some (1)	-1.26
job losses, threat of terrorism, property value decreasing, drug trafficing, becoming citizens and going on welfare	Extreme (2)	2.58
jobs crime taxes	Extreme (2)	0.11

Note: The table shows responses that were unanimously coded by the expert coders as belonging in categories 0, 1, or 2. The `SentimentIt` approach reveals significant variation within categories that is largely consistent with a close reading of the actual texts.

Table SI-4: Expert coder agreements (and disagreements) for survey responses

		Coder 2		
		None	Some	Extreme
Coder 1	None	90 (26%)	64 (18%)	14 (4%)
	Some	5 (1%)	89 (26%)	20 (6%)
	Extreme	1 (0%)	7 (2%)	50 (15%)

Figure SI-6: `SentimentIt` fearfulness estimates from separate runs



Note: We ran two rounds of twenty comparisons each on the fearfulness of survey responses regarding immigration one week apart. This plot shows the high correlation (0.86) and therefore high reliability between independent runs.

SI-1.4. *Supplemental information for analysis of State Department Reports*

One way to evaluate the quality of our codings is to closely examine cases where the crowd-sourced codings of the State Department reports differ from Hathaway scores (see Figure 5 in the main text). 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: 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). Our estimate of Sweden is also relatively high. Two paragraphs, discussing police using excessive force, drive this result. Hathaway does not consider isolated excessive force by police officers, if those officers were charged, to indicate torture. Relative to other paragraphs, however, these indicate Sweden was more of an edge case than the strict categorization suggests. Finally, our estimate for Benin is relatively high, but again one paragraph is driving this, and it states explicitly "that police sometimes beat criminal suspects," which is quite suggestive of torture.

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). It is not clear how this differs from the documents in Hathaway's first category. The estimate for Fiji is also relatively low. The only indication of torture in the document is the following line: "[p]olice sometimes abuse detainees; the authorities have punished some of the offending officers, but these punishments have not deterred all police abuses." In the coding scheme, a country should be coded as a one if beatings are from individual police who were subsequently punished.

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 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.[2]

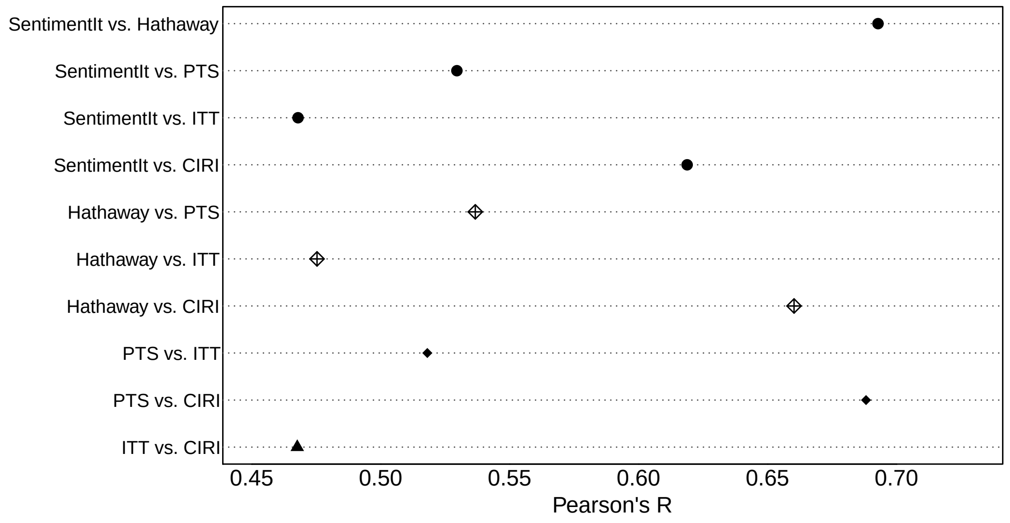
In the main text, we compare our measure with three other well-known measures of torture to assess the validity of these estimates as capturing the true degree of torture within the countries. First, we analyze the State Department variable of the Political Terror Scale (PTS) data on human rights violations (Gibney et al. 2015). This variable is constructed by a team of experienced coders, accounting for both the Human Rights Reports documents and other relevant 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 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-7 shows how all of these measures are correlated.

SI-2 HOW MANY COMPARISONS ARE NEEDED?

How many comparisons are needed to generate valid estimates of latent traits embedded in texts? To provide an answer to this question, we return to the movie review data discussed in Section SI-1.2. We estimated positivity measures using the first 10 pairwise comparisons (per doc-

[2]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.

Figure SI-7: 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.

ument), the first 20 comparisons, the first 30, and then the complete dataset. (Pairwise comparisons were constructed in blocks of 10 to make this analysis feasible.) Figure SI-8 shows the correlation between these estimates of positivity and the user-provided stars. After 10 comparisons per document, the correlation is 0.84. After 20, the correlation is 0.86. After 30 and 40, 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.

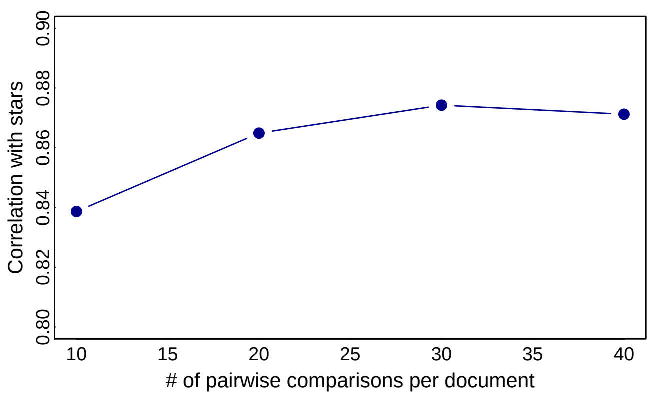
SI-3 EVALUATING WORKER QUALITY AND RELIABILITY

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-3.1. *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 re-

Figure SI-8: 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.

lated 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).

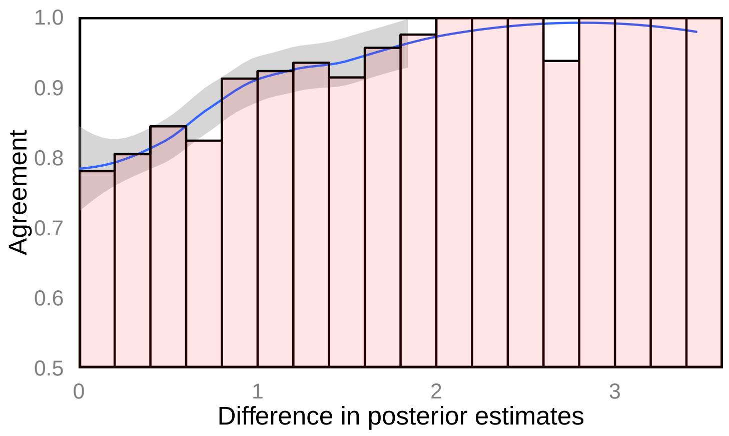
In the area of sentiment analysis, Hsueh, Melville, and Sindhwani (2009) compare AMT workers against expert coders to find that the aggregate accuracy of the online workforce approached the gold-standard set by the expert coders. This effect increased when removing the noisiest coders from analysis,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

SI-3.2. *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-1.1, 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-9 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-9: 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-3.3. *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-1.1 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-3.4. *No evidence of systematic worker biases*

One potential problem for our 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 of 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 in 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 mean 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-3.5. *Uncertainty in estimates*

One concern readers may have is that documents that score 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.[3]

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.

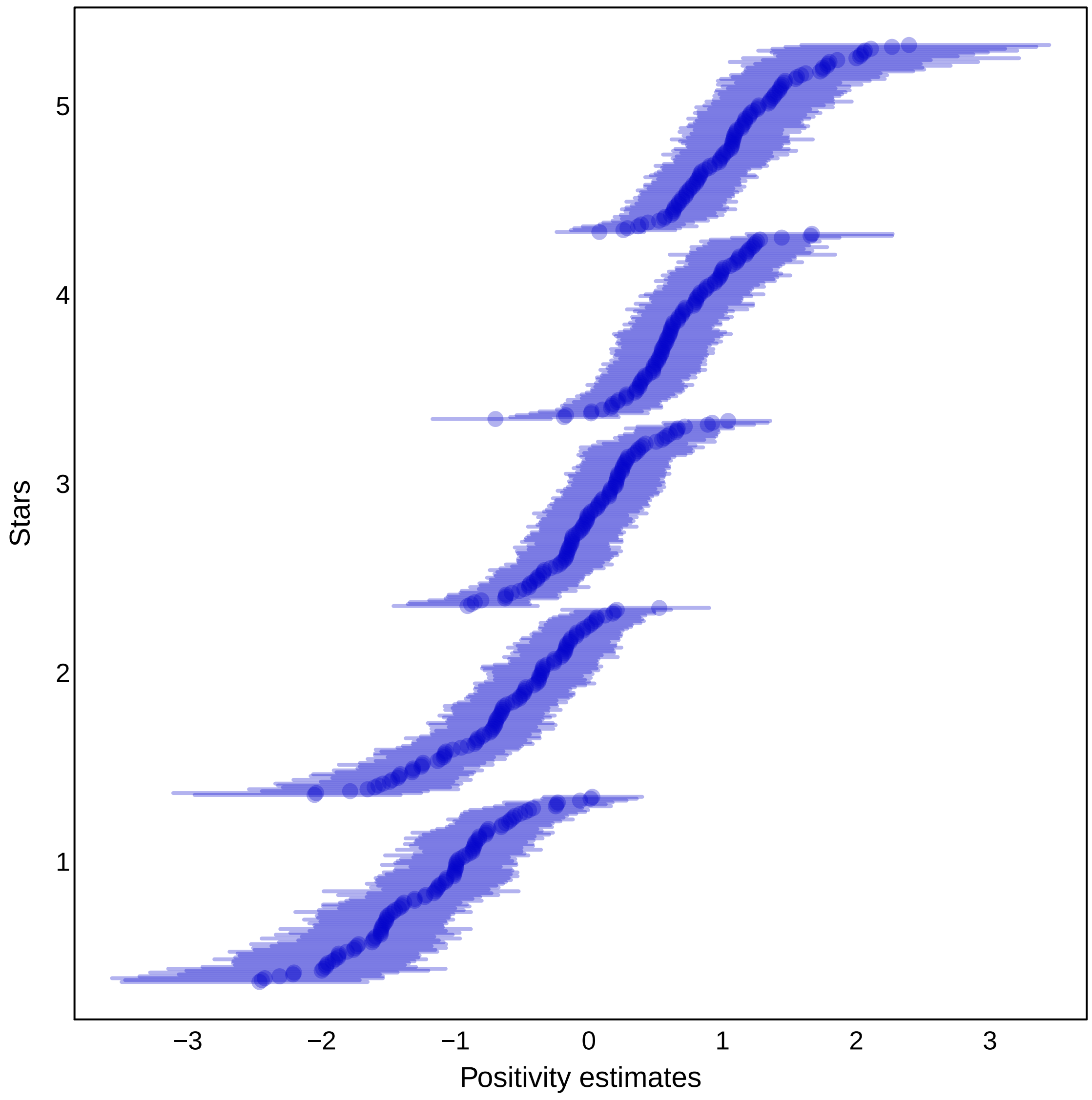
SI-3.6. *Completion times*

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-11 and SI-12 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).[4] As can be seen, absolutely no pattern emerges, indicating that speed of completion may be more related to the abilities and

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

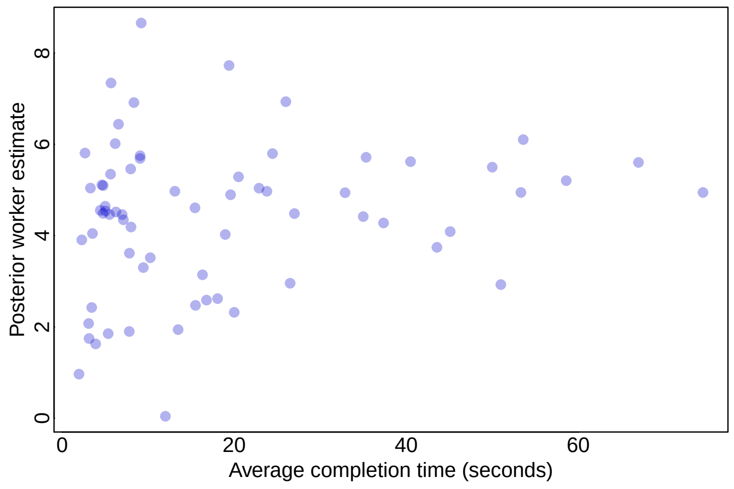
[4]Plots for the remaining applications look essentially identical.

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



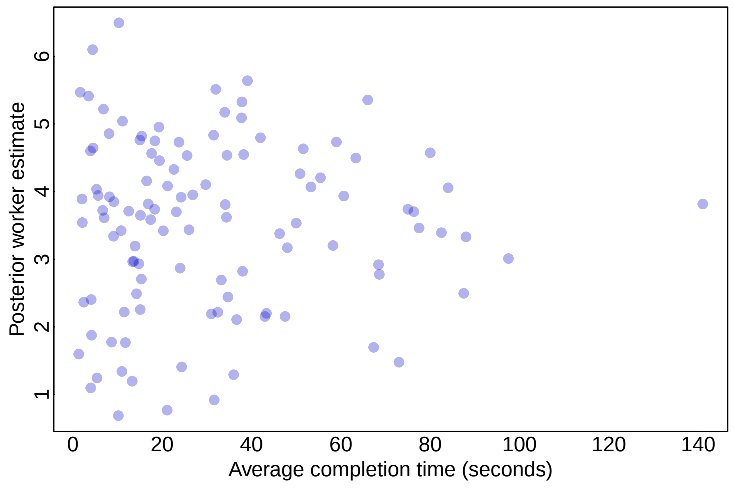
pacing of the worker than the quality of their evaluations.[5] Likewise, Figure SI-12 shows the variance of the posterior estimates for the workers on their average completion time in the movie review application. No strong relationship seems present, suggesting further that completion time is not a significant determinant in worker quality or our uncertainty of the estimates. Thus, although 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-11: Posterior worker estimates on average completion time: Movie review application



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

Figure SI-12: Posterior worker estimates on average completion time: Congressional advertisement application



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

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

Of course, as a helpful reviewer pointed out, response times themselves may not be dispositive as workers simply "spamming" the HITs may do so at a pace to prevent detection. Other methods must be relied upon to identify low quality workers. Another option would be to follow Benoit et al. (2016) in the addition of gold standard questions or attention screeners (Berinsky, Margolis, and Sances 2014). However, due to the pairwise comparisons framework, we are able to assess the quality of the data from each worker on an ongoing basis as new data become available. Specifically, we generate a parameter for each worker that indicates the degree to which their coding decisions are consonant with the decisions of the other workers. Since each comparison involves two documents, we require only a modest number of tasks to be completed before these parameters become diagnostic. However, while we believe that adding attention screeners is unnecessary, we note that with some simple programming, it would be straightforward to include them during the data collection process.

SI-4 COMPARING *SENTIMENTIT* TO AUTOMATED METHODS

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." In this Section, we briefly compare the performance of `SentimentIt` to several approaches for automated coding of texts.

Note that our purpose is **not** to show that these automated methods are "wrong." Indeed, we view one possible application of `SentimentIt` scores as providing labeled data for training machine learning algorithms. Instead, we aim only to provide limited evidence that our approach performs well relative to these increasingly popular methods and is thus not redundant.

SI-4.1. *Benchmarking against supervised learning*

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).[6] However, 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

[6]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).

ideology in texts against expert human coders and found that in many cases the statistical methods 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. 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. As a consequence, as Grimmer and Stewart (2013) discuss, "when dictionaries are created in one substantive area and then applied to another, serious errors can occur" (p. 274).

Thus, despite the promise of automated methods, in many cases relying on crowdsourced coding will provide superior estimates. To provide some sense of how our method compares to modern supervised learning methods, we compare our method with both `WordScore` and `WordFish`. Further, we compare our method to the approach developed by Socher et al. (2013)—one of the most successful and advanced approaches from the modern machine learning literature.

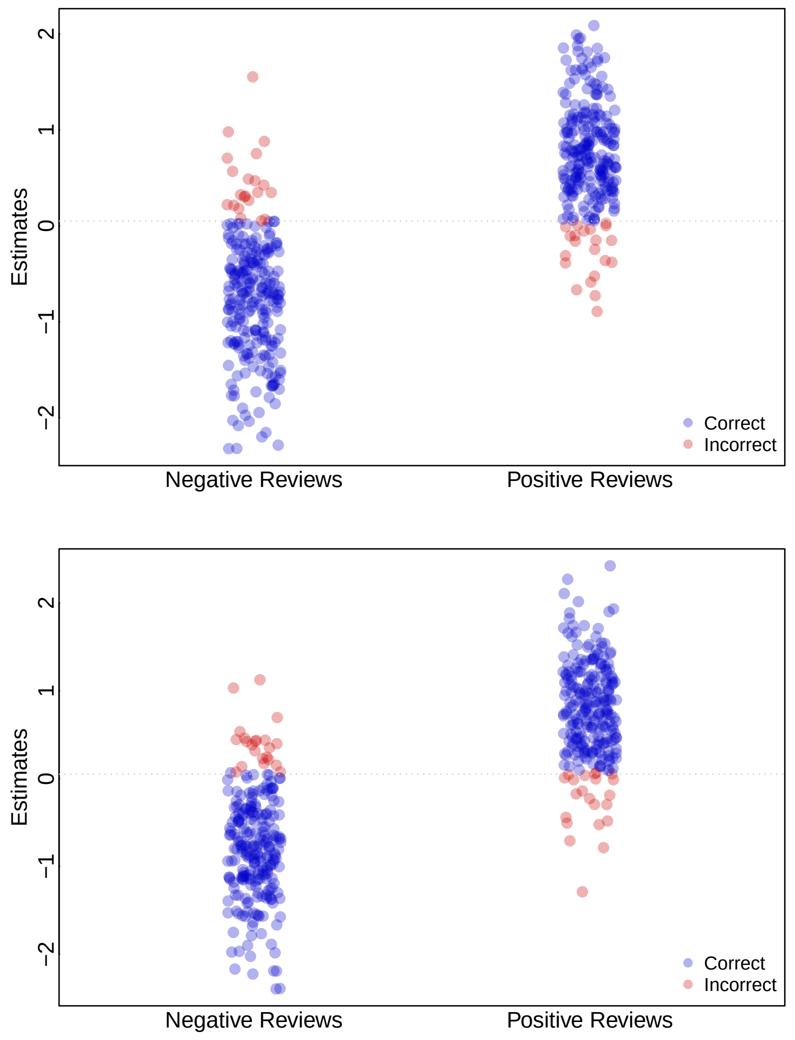
Pang and Lee (2005) analyze the polarity of 10,663 movie review snippets from Rotten Tomatoes. Half of the snippets (extracts from longer reviews) are taken from positive reviews, and half from negative reviews. Pang and Lee (2005) assume that snippets taken from positive reviews (e.g., 5-star reviews) 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 ran the experiment with and without a certification. We are able to classify 91.6% of the documents "correctly" without the qualification and 90.4% requiring a qualification. Figure SI-13 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.[7]

The performance of the supervised learning methods was noticeably worse. If we trained the `WordFish` algorithm on only the 500 snippets in our study, it classifies 53.4% of the documents

[7]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.

Figure SI-13: Identifying polarity in movie reviews without and with a qualification



Note: The top panel shows `SentimentIt` estimates without requiring a qualification and the bottom panel shows estimates requiring a qualification.

correctly. Training using the entire set of 10,605 documents, `WordFish` only correctly classifies 50.4% of the 500 documents correctly. Training on all of the snippets not in our test set (n=$10,105$), `WordScore` correctly classifies 76.8% of the remaining 500 snippets. Recalling that the Socher et al. (2013) method was only able to classify 85.4%, this evidence generally shows that the crowdsourced coding approach performs well relative to supervised learning algorithms.

SI-4.2. *Comparing to topic models*

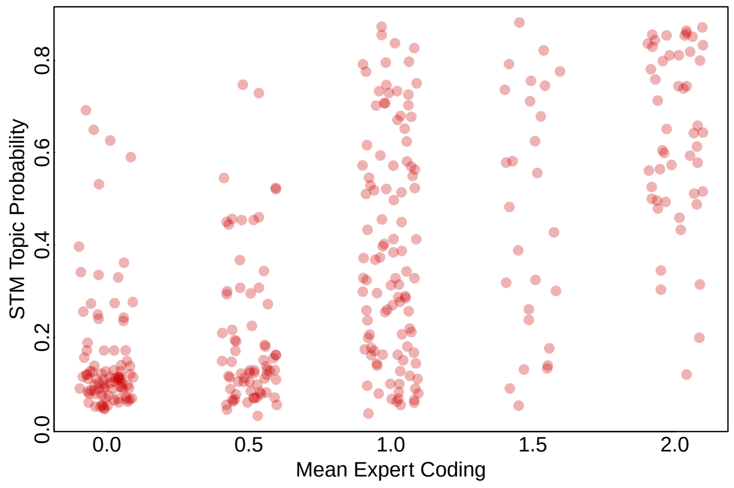
Topic models are not intended to provide measures of concepts within texts as defined *ex ante* by researchers. Nonetheless, it is increasingly common to see researchers using topic probabilities from these models as measures of meaningful latent traits within texts for quantitative analysis (e.g., Catalinac 2016). Thus, it is instructive to compare the performance of `SentimentIt` to this now-common approach to analyzing texts.

To do this, we return to the survey responses measuring fear discussed in Section SI-1.3. Roberts et al. (2014) analyzed these same statements using a structural topic model (STM), and identified one topic as indicating fearfulness. Figure SI-15 compares the correlations between `SentimentIt`, each expert coder, the and mean of the expert codes. Figure SI-14 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). In general, these topic probabilities do not capture the sense of anxiety or 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. In the end, Figure SI-15 shows that the STM probabilities are only correlated at 0.64 with the mean expert coding.

This degree to which topic models can imperfectly capture researcher-specified concepts is further illustrated by examining specific texts. Table SI-5 provides examples where the topic model most strongly disagrees with the expert coders. As can be seen, the topic model estimates several benign responses as fearful, such as "nothing" and "illegallanguage" (sic). The bottom entries are statements that the automated method 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 worries such as "changing the basic makeup of the United States."

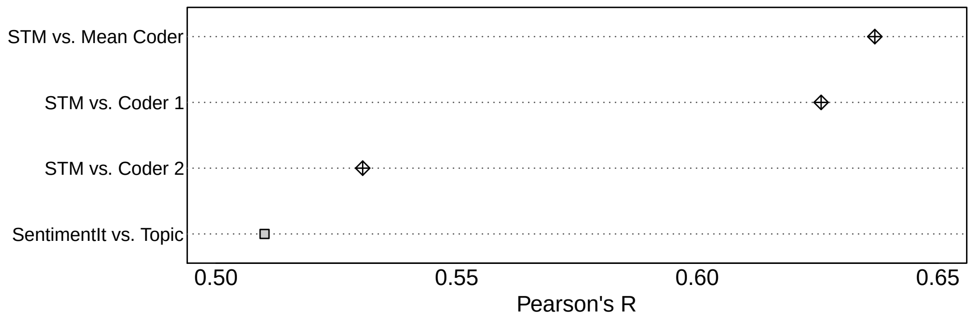
For completeness, we again compare this analysis to `WordFish` and `WordScore`. The correlation between the human coders and `WordFish` scores is very close to zero ($r = .02$). This is likely partially due to the sparse matrix and lack of overlap between (often misspelled) words in the corpus. To analyze the performance of `WordScore`, we train on half of the data using the human codes as the "truth," and test on the remaining half. The correlation between the scores and the human codes out of sample is 0.566.

Figure SI-14: Fearfulness estimates from `SentimentIt` and STM on expert coding



Note: The 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.

Figure SI-15: 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.

Table SI-5: Anomolous topic probability assignments among unanimously coded documents

Expert coders	Statement	STM prob. fearful	`SentimentIt`
None (0)	"nothing"	0.65	−2.20
None (0)	"they do not pay taxes"	0.69	−0.40
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	0.74
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	0.81

Note: When the structural topic model misclassifies, the topic model is more likely to be in error than the expert coders.

SI-5 ROBUSTNESS TO MODELING CHOICES

Table SI-6 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-6: Correlations between `SentimentIt` scores and mean coder selection

Application	Correlation
Campaign ads	0.96
Human rights reports	0.95
Movie reviews	0.96
Immigration survey	0.98

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-6 TRAINING AND CERTIFICATION

SI-6.1. *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.

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.

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.

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-6.2. *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.

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.

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 Data / Worker Certification. Here there will be a link "New." 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.

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 in order to disallow continuing, then 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 a task:

```
Qualtrics.SurveyEngine.addOnload(function()
{
this.disableNextButton();
var that = this;
var worker_id = "${q://QID29/ChoiceTextEntryValue/1}";
    //this is the worker ID inputted by the respondent at
    //the beginning of the survey (question ID 29, text field 1)
var post_url = "https://www.sentimentit.com/api/sessions.json";
$j.ajax({ //get auth token
type: 'POST',
url: post_url,
data: {email: 'name@school.edu', password: 'UniquePassword'},
    //replace with SentimentIt email and password
success: function (data) {
if(data.success == true){
auth_token = data.auth_token; //get the authorization token
}
},
});
function waitForElement () {
       //wait for the auth_token variable to be defined before executing next call
if(typeof auth_token !== "undefined"){
var get_url ="https://www.sentimentit.com/api/certifications/certname/turk_workers/" +
    worker_id + ".json?email=name@school.edu&auth_token="+auth_token;
        //replace with SentimentIt email and password, and change certname to the
        //banned certification associated with the certification, e.g. bannedcongress
$j.ajax({ //if the worker is banned, the next button will stay disabled
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) { //function will error if worker is not in system - means
        //the worker has not attempted any certifications so should
        //be allowed to continue
that.enableNextButton();
}
});
} else {
setTimeout(function(){
waitForElement();
},250);
}
}
waitForElement()
});
```

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:

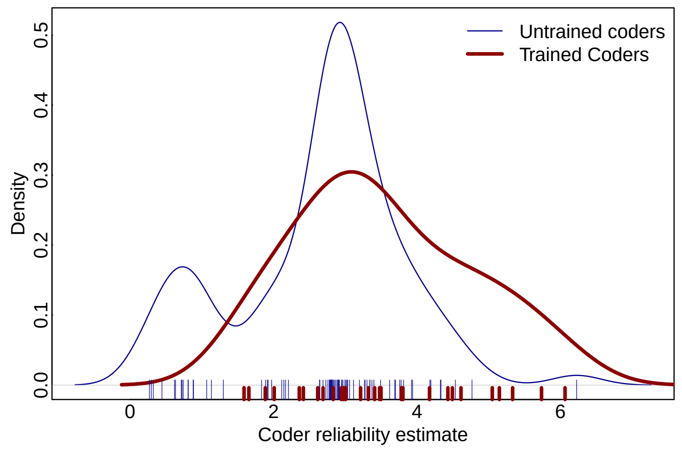
```
Qualtrics.SurveyEngine.addOnload(function()
{
var worker_id = "${q://QID29/ChoiceTextEntryValue/1}";
    //this is the worker ID inputted by the respondent at
    //the beginning of the survey (question ID 29, text field 1)
var post_url = "https://www.sentimentit.com/api/sessions.json";
$j.ajax({
type: 'POST',
url: post_url,
data: {email: 'name@school.edu', password: 'UniquePassword'},
    //replace with SentimentIt email and password
success: function (data) {
if(data.success == true){
auth_token = data.auth_token; //get the authorization token
}
},
});
function waitForElement () {
        //wait for the auth_token variable to be defined before executing next call
if(typeof auth_token !== "undefined"){
var input_data = "{\"email\":\"name@school.edu\",\"auth_token\":\"" + auth_token +
    "\",\"certification\": \"certname\",\"workers\":[\""+worker_id+"\"]}";
    //replace with SentimentIt email and password, and certname should
    //be the name of the passed certification in the block if the worker
    //passed the qualification, and should be the name
    //of the banned certification if the worker did not pass
$j.ajax({
type: 'POST',
url: "https://www.sentimentit.com/api/certifications/create.json",
contentType: 'application/json',

data: input_data,
success: function (data) {
console.log(data);
},
error: function (data) {
console.log(data);
},
processData: false,
});
} else {
setTimeout(function(){
waitForElement();
},250);
}
}
waitForElement ()
});
```

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-6.3. *Evidence on the advantage of training modules*

In Section SI-4.1, we briefly discussed an experiment where we coded snippets from movie reviews using untrained and trained workers. Figure SI-13 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-16 shows the worker-level estimates. Clearly, requiring a simple qualification test disincentivised some low-quality workers from participating.

Figure SI-16: 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-7 WORKING EXAMPLE

SI-7.1. *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.

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,sigma);
}
for(m in 1:M){
a[m] ~ normal(0,1);
}
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-7.2. *`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.

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.csv',
    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 example 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.csv",
    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_blLMfxcmlVOtOOF`

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 are 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 completing 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.

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 a full replication. 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-7 References

Alonso, Omar, and Ricardo Baeza-Yates. 2011. "Design and implementation of relevance assessments using crowdsourcing." In *Advances in information retrieval*, ed. Paul Clough, Colum Foley, Cathal Gurrin, Gareth J.F. Jones, Wessel Kraaij, Hyowon Lee, and Vanessa Mudoch. Springer.

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

Benoit, Kenneth, Drew Conway, Benjamin E. Lauderdale, Michael Laver, and Slava Mikhaylov. 2016. "Crowd-sourced text analysis: Reproducible and agile production of political data." *American Political Science Review* 110(2): 278–295.

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.

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.

Catalinac, Amy. 2016. "From Pork to Policy: The Rise of Programmatic Campaigning in Japanese Elections." *Journal of Politics* 78(1): 1–18.

Cingranelli, David L, David L Richards, and K Chad Clay. 2014. "The CIRI human rights dataset." `http://www.humanrightsdata.com`.

Conrad, Courtenay R., and Will H. Moore. 2012. "Ill-Treatment and Torture (ITT) Dataset." `http://www.politicalscience.uncc.edu/cconra16/UNCC/ITT_Data_Collection.html`.

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." `http://www.politicalterrorscale.org`.

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.

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.

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*.

Lauderdale, Benjamin, and Alexander Herzog. 2014. "Measuring political positions from legislative debate texts on heterogeneous topics." 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* 21(3): 267–297.

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

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.

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* 58(4): 1064–1082.

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.