Methodological Challenges in Estimating Tone: Application to News Coverage of the U.S. Economy

Machine learning methods have made possible the classification of large corpora of text by measures such as topic, tone, and ideology. However, even when using dictionary-based methods that require few inputs by the analyst beyond the text itself, many decisions must be made before a measure of any kind is produced from the text. When coding media the analyst must decide on the universe of media sources to sample from, as well as the criteria for selecting articles for coding from within that universe. If utilizing a supervised learning method, the method of generating training data presents many decisions: the unit of analysis to code, choice of coders, number of articles or units to code, number of coders per unit, and method of dealing with multiple codings of a single object. In this paper we consider the many decisions made by the analyst in using machine learning to classify media texts—using as a running example efforts to measure the tone (positive, negative, neutral) of newspaper coverage of the economy—and highlight our key findings throughout. In particular, we show that the decision of how to choose the corpus matters a great deal. We also introduce coder variance as a simple but novel measure of coder quality, and we demonstrate that this concept can be used to illustrate the varying returns to using multiple coders versus larger sample sizes in construction of a training dataset optimized for best classifier production. Finally, we introduce Classifer Training Using Multiple Codings, an improved method of utilizing multiple codings of individual objects, and demonstrate through simulation that it outperforms alternatives.

August 26, 2016

Pablo Barberá
University of Southern California

Amber Boydstun
University of California, Davis

Suzanna Linn
Penn State University

Ryan McMahon
Penn State University

Jonathan Nagler
New York University

Pablo Barberá is Assistant Professor of International Relations at USC (pbarbera@usc.edu). Amber Boydstun is Associate Professor of Political Science at the University of California, Davis (aboydstun@ucdavis.edu). Suzanna Linn is Professor of Political Science at Penn State University (slinn@la.psu.edu). Ryan McMahon is a PhD candidate at Penn State University (rbm166@psu.edu). Jonathan Nagler is Professor of Politics at New York University and a Principal Investigator in the NYU Social Media and Political Participation Laboratory (jonathan.nagler@nyu.edu). Authors are listed in alphabetical order. This material is based in part on work supported by the National Science Foundation under IGERT Grant DGE-1144860, Big Data Social Science, and funding provided by the Moore and Sloan Foundations. Nagler's work was supported in part by a grant from the INSPIRE program of the National Science Foundation (Award SES-1248077).

The analysis of text is at the center of a large and growing number of research questions in the social sciences (Grimmer & Stewart 2013). While analysts have long been interested in the tone and content of such things as media coverage of the economy (Soroka, Stecula & Wlezien 2015, Young & Soroka 2012, Goidel et al. 2010, De Boef & Kellstedt 2004, Tetlock 2007, Doms & Morin 2004), congressional bills (Jurka et al. 2013, Hillard, Purpura & Wilkerson 2008), party positions (Grimmer, Messing & Westwood 2012, Monroe, Colaresi & Quinn 2008, Laver, Benoit & Garry 2003), and presidential campaigns (Eshbaugh-Soha 2010), the advent of machine learning methods combined with the broad reach of digital text archives has led to an explosion in the scope and extent of textual analysis. Whereas researchers were once limited to analyses based on text that was read and hand-coded by humans, dictionary and supervised machine learning tools are now the norm (Grimmer & Stewart 2013). The time and cost of the analysis of text has thus dropped precipitously. But the use of automated methods of text analysis requires making a number of decisions that are often given little consideration and have consequences that are not obvious or benign. In this paper we present the issues raised by a number of decisions, demonstrate their consequences, and offer guidelines for thinking about these decisions in applied work when the coding of text is driven by machine learning algorithims.

The discussion in this paper is limited to coding the *tone* of text, rather than other variables such as topics or events, and specifically the tone of coverage of the U.S. national economy as presented to the American public in the *New York Times*.[1] But much of what we present is applicable to the analysis of text more broadly, both when using a computational approach and even (in the first stages we discuss below) when using manual content analysis. We lay out five decisions the analyst must make in order to produce a measure of tone. The analyst must: 1) decide the universe of text; 2) select a method to identify a population corpus from the universe; 3) choose whether to use a dictionary method or a machine learning method to classify each text in the corpus; and in cases of supervised machine learning 4)

[1] Tone is also termed ‘sentiment’ in the computer science literature.

decide how to produce the training dataset—the unit of analysis, the selection of coders, the number of objects to code, and the number of coders to assign to each object; and 5) decide how to optimally use multiple codings for classification. We present evidence as to the consequences of a number of choices open to the analyst at each step in this decision-making process and offer guidelines for how to make these decisions. Our primary argument is that *all* of these decisions matter and must be confronted in order to produce a valid measure of tone (or any other variable, for that matter).

Some of the results and advice we present here may seem self-evident. If one chooses the wrong corpus of media to code, no coding scheme will accurately capture the 'truth' of what the media is presenting. But in addition we show that less obvious decisions also matter. In our running empirical example in the paper—the coding of media tone of the U.S. national economy—we show that two reasonable attempts to select a corpus from identical media sources can produce very different outcomes. We also demonstrate that supervised machine learning out-performs dictionary methods, even in cases where a specialized dictionary is available. We introduce a simple but novel measure of coder quality (coder variance), and we demonstrate that this concept can be used to illustrate the varying returns of using multiple coders versus larger sample sizes in the construction of a training dataset optimized for best classifier production. It also provides useful guidance to analysts in evaluating different groups of coders, as well as in deciding how many coders to utilize. We also introduce, and demonstrate through simulation, an improved method of utilizing multiple codings of individual objects (Classifier Training using Multiple Codings) in creating a classifier that considers the variance in the codings to increase the statistical power of the training dataset.

Before proceeding to describe the decisions at hand, we note a key assumption associated with measuring tone: the assumption that there is a single, 'true' tone inherent in the text that has merely to be extracted. Of course that assumption is probably false. In the case of the tone of media coverage of the economy, different readers of the same text may interpret the story differently, whether because of different levels of economic sophistication or different economic interests. A layperson may not learn anything about the state of the national economy from an article on the bond market, whereas a sophisticated investor might learn a great deal. And a poor person owning no stocks may feel that stories about the stock market are not nearly as informative about the state of the national economy as are stories about the unemployment rate. However, we will proceed under the assumptions that: a) there is a true tone and b) the media reports a signal that is—on average—interpreted by all people in the same way.

1 Decisions the Analyst Must Make

The analyst preparing to conduct some automated method of analyzing text must make a number of decisions, each of which affects the ultimate measure of tone. Here we take up these questions in the order they present themselves to the analyst. We discuss the theoretical issues at hand and how we will assess the trade-offs associated with each. In the next section we present evidence regarding the impact of these decisions in the context of coding the tone of coverage of the U.S. national economy using the *New York Times* and discuss general guidelines for the analyst.

1.1 Issues in Selecting the Universe of Text

The first decision the analyst must make is how to identify the relevant universe of text. This is necessarily a problem-specific question. If one wants to know the tone of legislative speech in Canada, for example, the universe of text is well-defined (the written record of all legislative speech) and the amount of text is (relatively) small. But when the research question is about legislative speech more broadly, the appropriate universe of text is less obvious. When the research question is to know the tone of a topic presented to the public by 'media,' the scope is so broad as to be (at least given current archives and technology) unmanageable for most projects. The analyst is unlikely to code media sentiment by coding the entire universe of news stories—that is, *every* article printed in *every* newspaper, and *every* story aired on television, to say nothing of radio and social media, etc., everywhere in the world. For both practical (time and money) as well as empirical (introducing noise) reasons, the analyst is likely to choose some subset of media from which to sample. This choice likely matters a lot; sentiment expressed in different types of media is likely to be distinct.

For the purpose of comparison with published work and to illustrate the impact of different decision rules on measures of tone, we set aside this important question and define our universe of media to be articles from the *New York Times.* We focus our attention on the different ways we can draw a sample of articles about the economy from our population of interest and the consequences of different decisions for measurement of tone.

Before proceeding we define the terms we will use throughout. The **Universe** is the media source from which the analyst chooses to draw text. The **Population** is the set of articles in the universe that is relevant to the topic the analyst wishes to analyze. The **Sample** will be the sample of articles from the population that the analyst selects to use to measure the population. Note that barring omniscience—the analyst is not able ex-ante to identify the entire population of articles relevant to any topic. Thus it is not an option to choose a sample to match a population.[2] In fact, the **Sample** will contain articles from the population, *and* articles that are not in the population - articles the analyst believes ex-ante to be relevant based on some selection criteria, but which are actually not relevant. Finally, in cases where supervised machine learning is utilized, we refer to the *Training Sample* as the set of articles that will be human-coded and then used to train a classifier.

[2] This distinction would disappear if the analyst's research question is in knowing the tone of all articles that use the word 'economy'; but that research question is divorced from the question of how the media actually presents information about the economy.

1.2 Issues in Selecting the Sample of Text (i.e., the Corpus)

Deciding the universe of text (e.g., news source) is the first step in selecting the corpus to be analyzed. Given the universe, the analyst must identify the population of news articles, speeches, etc., that are of interest (e.g., that are about the U.S. economy). Any strategy to identify the population of interest has trade-offs. Any strategy that results in a sample that includes many irrelevant texts would add noise to the measure of tone, potentially swamping the signal, as texts would be treated as revealing sentiment that are not on topic. It would also add cost to the production of a usable training dataset for machine classification. At the other extreme, any strategy that excludes relevant text would mean at best having a noisier measure of sentiment: we would be unnecessarily decreasing the size of our corpus, and thus induce more sampling variation into our measure. At worst, such a strategy could introduce bias by omitting a set of relevant texts with sentiment distinct from the text included.

When identifying the population of text from archives of newspaper articles, the analyst has two main options.[3] She can rely on keyword (or regular expression) searches, or she can rely on some institutionalized, and generally proprietary, categorization of stories provided by the media entity (or catalogue of the media entity). Consider the *New York Times*. The paper archive is available via Lexis-Nexis Academic Universe as well as ProQuest. We can produce a keyword search that would retrieve all articles from the *New York Times* containing a given set of keywords aimed at identifying our subject of interest, and/or we could retrieve articles that appear to contain our subject by using the media provider's list of pre-defined subject categories. However, the problem with using the subject categories provided by the media provider (or by the media entity itself) is that they are both non-transparent (the methods and algorithms used to categorize news articles into these categories are proprietary) and, thus, non-transferrable (i.e., impossible to replicate in other contexts). Furthermore, the classification rules used by the provider to categorize articles by

[3] Note that these two options exist in many cases of choosing text: they merely require that some other entity has tagged or catagorized the text.

subject categories may (and do) change over time—indeed, even the list of subject categories available changes over time—with the consequence that one cannot replicate their use across other media or determine how broad or narrow the search really is. In other words, the same search performed using the provider's subject categories at two different points in time may yield very different results. The lack of transparency is a huge problem for scientific research; the analyst is completely dependent on the media provider to have accurately categorized articles about the economy. For these reasons, we recommend against using the media (or provider) subject classifications.

In fact, for most purposes we can not recommend strongly enough against using media provided categories. There is no way to transport them across publications, even within countries. In contrast, keyword searches are easily transported across countries via straightforward translation. Also, there is no way to know if the search criteria have changed over time. If the media provider is using human coders, then changing coders could cause a change in content independent of actual content. And this is invisible to the analyst. There are many reasons any keyword search can be problematic (relevant terms can change over time, different publications can use overlooked synonyms, and so on)– but these are all *transparent and known to the analyst and known to future analysts!* One *potential* positive aspect to using subject categories is that if articles were classified by humans, it is possible that the person who did them carefully chose appropriate categories at different time periods. But of course this is also possible with keyword searches.

A boolean regular expression search gives the analyst flexibility to be either broad or narrow in choosing the keywords (or key terms) for her search. It is straightforward to examine the quantity of articles that different searches return and the search is replicable across papers and media providers. Of course if the analyst chooses to do a keyword search, the choice of keywords becomes 'key'.[4,5] And because the underlying set of archived articles

[4] See King, Lam, and Roberts (2016) and cites therein for a discussion of methods of keyword generation.

[5] We note that regular expression searches are not necessarily restricted by the media provider's front-end. One can of course download articles more broadly than one intends to code, then after downloading,

can vary slightly over time, based on the media provider's contracts with news outlets and the provider's internal archiving parameters, even the same keyword search performed at two points in time may yield different results, although the differences should be less than those suffered using subject categories (Fan, Geddes & Flory 2013).

Our goal is to assess the effect of different search strategies on the corpus itself and the resulting measure of tone (of the U.S. national economy) when the specific universe of text is comprised of news articles from the *New York Times*. In what follows we ask: Do different, sensible strategies to identify the population of articles produce qualitatively different corpora? How much is the resulting measure of tone affected by these decisions? After describing our endeavors to tackle these questions, we will offer suggestions for "best practices" strategies other analysts might consider when identifying the corpus from a universe of news articles.

1.3 Issues Concerning Dictionaries and Supervised Machine Learning Classifiers

Once the analyst identifies a corpus, there are two fundamental options for coding sentiment (beyond traditional manual content analysis): dictionary methods and supervised machine learning methods. There are, as with all methods, costs and benefits to both of these approaches.

1.3.1 Dictionaries

A dictionary approach to coding tone entails defining a set of rules that are then applied to the corpus to generate the measure of interest. Typically the use of dictionaries involves identifying document features (generally words or short phrases) as positive or negative, counting the occurrence of each, and assigning a weight to each feature (usually +1 for

post-filter based on search criteria of any given complexity.

positive and -1 for negative). For any given unit of analysis the weighted term values are added and normed relative to the total number of words in the text. The result becomes the tone 'score' for that unit of text.

If the analyst chooses to code text using a dictionary, she must either create a dictionary or use an off-the-shelf dictionary, such as Lexicoder (Young & Soroka 2012) or SentiStrength (Thelwall et al. 2010). Creating a dictionary presents its own set of challenges, such as defining the rules used for coding and how they will be applied to the text. Appropriating a previously created sentiment dictionary has the obvious advantage of providing an inexpensive method for the analysis of tone. Given a dictionary of terms—either a subject-specific set designed for the topic at hand, or a general set of terms—one can simply process the text at essentially no cost. This is not to say that the application of dictionary methods is researcher independent. Once a dictionary has been created or selected there are still decisions to be made by the analyst concerning the processing, or cleaning, of the text and the unit of analysis to code.

Although it is the most cost efficient method for classifying text, there are many limitations inherent in relying on dictionaries. The first critique of a dictionary approach is that it ignores the context in which features (e.g., words) appear. A news report that "the economy is doing *better*," for example, is the exact opposite of a news report that "the economy is not doing *better*," and yet most dictionaries would treat both sentences as the same, including a single positive term.

A second critique is that sentiment dictionaries are highly domain specific; it is impossible to generate a universally applicable dictionary. A dictionary for conducting sentiment analysis of social media, for example, is not going to translate to a content analysis of news media. In an analysis of corporate 10-K reports, for example, Loughran and McDonald find that "... almost three-fourths (73.8%) of the negative word counts according to the Harvard [IV Dictionary] list are attributable to words that are typically not negative in a financial context," (2011, 36). Although the dictionary and its application were both related to sentiment, they were clearly not compatible. In practice, this means that selecting which of the many potential dictionaries is best suited to the analyst's task is arguably the most important decision that is made when dictionary methods are applied.

A third, independent, limitation of dictionaries is that they unnecessarily constrain the feature space to some subjective set of words. Even if the analyst creates or selects an appropriate dictionary, there will be features that are likely relevant to the analysis, but are missing from the dictionary. By defining the set of 'relevant' features a priori, the analyst is severely limiting the amount of information they can use from the text.[6] Related to this issue, the weight attached to each feature is assumed known ex-ante. In the canonical case all words are treated as equally informative with positive features weighted by +1 and negative features by -1; though some dictionaries (including SentiStrength) relax the limitation of +1/-1 scores (Eshbaugh-Soha 2010, Soroka, Stecula & Wlezien 2015).

To see how restrictive this combination of assumptions is, consider the following thought-experiment. Take a set of articles, or sentences, about the economy and ask humans to rate them on a scale of 1 to 9 for how negative or positive they are about the economy. Then run a regression of the scores for the article against each word in each article, with an appropriate penalty function. What are the chances that the set of words in a dictionary such as Lexicoder (Young & Soroka 2012) turn up as the words with coefficients of 1.0 on the righthand side, and that all words omitted from the dictionary have coefficient 0? Of course we don't have to make this a thought experiment—we performed the actual experiment. While technical differences in treatment of words make direct comparisons difficult—it is of course not surprising to report that the estimated coefficients for terms included in Lexicoder varied widely. It is perhaps surprising to report that barely 20% of the terms we estimated to be most important according to the magnitude of their regression coefficients appear in

[6] This issue is exacerbated when the analyst wishes to examine any unit of text larger than the usual unigrams (or negated unigrams). If the dictionary is expanded to include pairs or trios of words, the number of potential features increases very quickly and adequate feature selection becomes untenable.

Lexicoder.

Finally, results from dictionary methods are difficult to validate because they are independent of any actual human input on the document level. The analyst could have human coders examine documents and review how the dictionary is performing, but at that point the cost benefits of using a dictionary begin to deteriorate.[7]

1.3.2 Supervised Machine Learning

The analyst selecting supervised machine learning methods to analyze the tone of the text follows three broad steps. First, a sample of the text (the training dataset) is coded by humans for tone (i.e., the text is labeled). Then a classification method is selected, and trained to predict the label assigned by the coders within the training dataset.[8] In this way the classifier learns the relevant features of the dataset and the weight assigned to each. Multiple classification methods are generally applied to the data and tested for minimum levels of accuracy using cross-validation to determine the best classifier. Finally, the chosen classifier is applied to the entire corpus of text being analyzed (the 'Sample' in our terms) to predict the sentiment of all other unclassified articles.

The primary downside of supervised machine learning methods is that they are highly labor intensive. They require the production of a training dataset, which requires human coders. The creation of the training dataset also requires the analyst to make a number of decisions. These decisions involve such things as the unit of analysis, who will code the data, the number of coders, and number of objects to be coded, as well as decisions about preprocessing the text (i.e. handling stemming and stop words).[9] As we show below,

[7] Analysts often purport to validate results using some other variables the analyst expects are related. This 'convergent validity' approach, not without problems, estimates the extent to which two measures that should, in theory, be related are, in fact, empirically related. This cannot, however, guarantee that the method is capturing what the analyst is attempting to measure and brings with it the obvious problem that the analyst is then trying to simultaneously test her *theory* of what tone is related to, *and* her measure of tone.

[8] In traditional econometric terms: a model specification is selected, and model parameters are estimated.

[9] Some of these decisions are also required when using dictionary methods, such as the unit of analysis,

these decisions matter for the measure of tone produced. Other issues may also arise. For example, does the analyst need to train a separate classifier for different time periods or different media sources? And depending on the type of corpus, sometimes even the best classification algorithms cannot reach minimally acceptable levels of accuracy for replicating human coding of tone.[10] An additional drawback is that, as with a dictionary approach, the quality of machine learning methods can vary, depending on the task and corpus at hand. An algorithm that works very well in replicating human coding for the *New York Times*, for example, may perform poorly for *USA Today*.

Yet despite these drawbacks, there are many advantages to using supervised machine learning over dictionaries. First, we don't need to worry about identifying the relevant feature set a priori. Nor do we have to make arbitrary decisions about the weight to attach to specific features of the text. Instead the relevant features and their weights are estimated from the data. The feature space is thus likely to be much larger and thus more comprehensive than that used in a dictionary. Supervised machine learning can also allow the word context to matter, by use of n-grams or co-occurences as features. Finally, the use of supervised machine learning methods produces a measure of tone that can be evaluated with measures of accuracy and precision using cross-validation.

In the analyses that follow we will compare the results from two dictionaries—Lexicoder (Young & Soroka 2012) and Sentistrength (Thelwall et al. 2010)—and Hopkins 9-Word Method (Hopkins 2010),[11] to results from supervised machine learning methods. We show that a supervised machine learning classifier can outperform even a task-specific dictionary such as Lexicoder based on accuracy of classification.

and how to preprocess the text.

[10] One advantage of supervised machine learning is that the analyst *knows* the level of accuracy the classifier reaches. With dictionary methods, the analyst produces a measure with no measure of precision or uncertainty.

[11] Hopkins' method consists of counting the number of articles per month mentioning nine economic words (inflat, recess, unempl, slump, layoff, jobless, invest, grow, growth).

1.4 Making Decisions about the Training Dataset: How do we select the unit of analysis, the coders, number of coders and number of objects to code?

To use supervised machine learning to train a classifier the analyst must make several decisions when creating a training dataset.[12] The analyst must: a) choose a level of analysis; b) choose a source of coders; c) decide how many coders to employ; and d) determine the number of objects to be coded for the training dataset.[13]

To make these decisions we need to consider the purpose of the training data. We are going to use the training dataset to train a classifier, i.e., we are going to estimate a model based on features of the objects coded where the training dataset consists of the outcome variable (the dependent variable) that is provided by human coders, and the text of the objects. We want to develop a model that best predicts the outcome (Y) out of sample. We know that to get the best possible estimates of the parameters of the model we are concerned with measurement error about Y in our sample, the size of our sample, and the variance about our independent variables. Since as we see below measurement error about Y will be a function of the quality of coders and the number of coders we use per object, it is impossible to consider number of coders and size of the training set independently. Given the obvious existence of some budget constraint we will need to make a choice between more coders per object, and more objects coded. Additionally, the question of optimal unit of analysis needs to be addressed. For a given budget constraint, more text can be coded if it is coded at a more highly aggregated level (e.g., articles versus paragraphs). But there may be cost in noise generated in the data that makes precise estimates of parameters harder to achieve. Below we discuss the theoretical trade-offs associated with these decisions.

[12] The terminology of ‘training a classifier’ is peculiar to machine learning, but easily translates to traditional econometrics as: choose the model specification, and estimate model parameters.

[13] See the appendix for a discussion of how to develop a coding instrument.

1.4.1 Issues in Selecting a Unit of Analysis

One of the first decisions the analyst must make when deciding how to produce a training dataset is the unit of analysis to code. Should a supervised machine learning classifier be trained on sentences, paragraphs or entire articles?[14] We expect that coding at the sentence level should produce more *precise* (less noisy) information about the relationship between text features and tone, though this will depend upon the distribution of features. If sentences typically contain features that are (nearly) all positively or negatively associated with tone while articles contain a mix of positive and negative features, then the precision of coding at the sentence level will be quite high compared to that from article level coding and we might expect better results from sentence level coding. However, there is uncertainty over how coders will perform on sentence versus articles.

In order to determine the optimal unit of analysis in coding the tone of economic news coverage in the *New York Times*, we compare the out-of-sample predictive accuracy of classifiers (at the article level) produced from sentence and article coding and then, again, offer suggestions for how other analysts might approach this trade-off.

1.4.2 Choice of Coders

There are at least three possible sources of coders to use to create a training dataset. First, the analysts themselves, presumably experts in the topic at hand, could code the data. Second, the analyst could rely on a pool of coders who are jointly trained for the task

[14] Technically, this decision must be made when using dictionaries as well but the issue is distinct. The issue for dictionaries is the level of specificity used in aggregating the information for the purposes of determining the tone of the article. If some sentences contained more positive words and others more negative words and the information is used to characterize the overall tone of the sentence, which is then aggregated by calculating the proportion of positive sentences, the result might be quite different than simply calculating the proportion of positive features in the entire article. The same aggregation issues applies to machine learning classifiers. Ultimately we are interesting in creating a measure of tone of media coverage in a given month. This might suggest we code the full sample of text in the month to produce a single measure. The norm, for both dictionary and machine learning classifiers, is to code articles, which are then averaged in some way to produce a monthly measure of tone.

at hand. In practice, this would mean undergraduate or graduate students trained to do the task. Third, the task could be outsourced to a broader *untrained* labor pool using a labor contracting pool such as CrowdFlower or Amazon's MTurk. Relying on the analysts themselves (ourselves as the case may be) limits the labor pool, and thus limits the size and scope of the training dataset. Using students who are trained by the analysts can maximize inter-coder reliability if the students are trained as a group but is relatively expensive and time-consuming. Outsourcing the task can quicken the speed of the coding, as the labor pool is potentially quite large, and reduce its total cost.

We present one way to evaluate the quality of coders that, as we will see, also helps inform decisions about the trade-off between number of coders and number of objects coded. Assume that each coder produces an unbiased estimate of the object to be coded. Then any given coding by coder J about some object i will be the truth (θ_i) plus some error (ϵ_{ij}). So given the unbiasedness of coders assumption, we can define the 'quality' of coders in terms of the average amount of error in their codings, i.e., the variance of their codings. So any coder j is defined by the variance of her codings, σ^2_j. Note that since we have assumed unbiasedness, this is assumed to be variance about the true tone of the object. We assume that our objects, i.e., articles, will differ in the amount of error they generate when coding. In other words, some objects will be more ambiguous and thus harder to code. We can refine our description of each coder to say that she has some variance in her coding about each object i, $\sigma^2_{j,i}$, and for two different objects, k and l, we believe that:

$$\sigma^2_{j,k} \neq \sigma^2_{j,l} \; \forall \; k, l. \tag{1}$$

If we are concerned with the mean estimate of tone for an object i ($\bar{\hat{Y_i}}$), and if the errors coders make are independent, then the variance in our estimate of the truth drops as we add coders.

Now assume we have two distinct sets of coders, untrained crowdsourced coders from

CrowdFlower (CF), and trained undergraduates (UG). We can characterize the quality of the coders by the average variance of a single coder from each group over our set of objects: σ^2_{cf} and σ^2_{ug}. Now if our goal is to get the highest possible quality estimate of each object i, then we are not interested in inter-coder *agreement* about the coding of the object – the agreement could include 2 coders both agreeing on a coding that is an error. Note that traditional measures of inter-coder reliability are actually measures of *agreement.* Those measures are useful for validating a coding scheme and understanding level of *agreement*, but if we are interested in having a training dataset where we reduce root mean squared error about truth, then thinking of variance about a mean rather than *agreement* is what we want to do.[15][16]

So, we have a straightforward exercise. Should we use undergraduates, or CrowdFlower coders? And if so, how many? Say that the average variance of CrowdFlower coders over all of our objects to be coded is σ^2_{cf}, and the average variance of Undergraduate coders over all of our objects to be coded is σ^2_{ug}. We assume that σ^2_{cf} is higher than σ^2_{ug}, this is likely as the undergraduates have been trained together and the training is designed to minimize inter-coder differences. The undergraduates are also a more homogeneous group than the CrowdFlower coders. Given this, if we are going to choose the mean coding of an object i as the value of the object, then in order to get the dataset with minimum root mean squared error, the choice between undergraduate coders and CrowdFlower coders is based on the variance of the *mean* coding of the object i. If we have J_1 undergraduate coders, and J_2 CrowdFlower coders then we know that the variance of the mean coding over all objects will be σ^2_{ug}/J_1 for undergraduates, and σ^2_{cf}/J_2 for CrowdFlower coders. This is our key insight here: we do not care about comparing σ^2_{cf} to σ^2_{ug}, but rather we care about comparing (σ^2_{ug}/J_1) to (σ^2_{cf}/J_2). Or, another way to realize this is that we care not about the error from

[15] If we have a coding scheme that is binary, then talking about the variance of coding and maintaining our unbiasedness assumption would be less tenable. However, we use two different coding schemes here: one a five-point scheme, and the other a 9-point scheme. The majority of estimates in the paper are based on the 9 point coding scheme.

[16] See Grimmer, King and Superti for a longer discussion of the problem of inter-coder reliability (2015).

a single coder $(Y_i - \widehat{Y_{i,j}})$ but we care about the error from the *mean* coding of an object: $(Y_i - \bar{\hat{Y}}_i)$.

Say that each undergraduate coding costs z times as much as each CrowdFlower coding. This suggests that if we have a fixed budget constraint, $J_2 = z \times J_1$. What we are now comparing is σ^2_{ug}/J_1 to $\sigma^2_{cf}/(z \times J_1)$. Then if we treat the sample size of articles to be coded as fixed (i.e., that either set of coders would be coding the same set of articles), and if we are simply choosing based on cost, we pick CrowdFlower coders if:

$$\frac{\sigma^2_{cf}}{\sigma^2_{ug}} < z \tag{2}$$

In other words, if the ratio of the variance of coding of CrowdFlower coders to the variance of coding of undergraduate coders is less than the ratio of the cost of the two coders, then we should choose the CrowdFlower coders as the higher number of less precise CrowdFlower codings will lead to a more precise estimate than the smaller number of more precise undergraduate codings. Thus this choice is easy to make provided the analyst has estimates on the two variance parameters. The analyst could get estimates on those parameters by simply having some sample of objects coded by multiple coders from each of the two groups. And as we describe below, knowing the variance of the coders is crucial to another (joint) decision: the number of articles to code and how many coders to use.

We assess the quality of Penn State undergraduate coders vis a vis Crowdflower workers by applying our variance measure to a subset of sentences coded by each group and a broader set of codings within each group.

1.4.3 Selecting Number of Coders and Number of Objects Coded

The discussion above implies the choice of who to code, how many coders and the *number of objects to code* should in fact be made jointly by the analyst. The relevant question is not only between CrowdFlower or undergraduate coders, but also between *more objects coded with fewer coders* versus *fewer objects coded with more coders*. This tradeoff will obviously depend on the variance of coders. In the limiting case, if variance for coders is 0 (every coder gives the correct coding for each object), then there is zero value to additional coders.

As we described above, the objects we are coding are to be used in a training dataset and this means that they are the dependent variable in some sort of model. Thus any error about the coding of the object is measurement error about the dependent variable (Y) when we estimate the model. So the problem for the analyst is how to choose the optimal number of coders per object and number of objects coded given the quality of coders. In linear models at least, we know that both those quantities (number of observations, and the average measurement error) figure into the precision of our estimates in a predictable way. In models typically used in supervised machine learning, the exact relationship between those quantities is not necessarily as clear. In results presented later in this paper we show how accuracy of a classifier changes with different combinations of number of objects coded and number of coders in efforts to create the best measure of tone of media coverage of the U.S. national economy.

1.5 How to Efficiently Treat Multiple Codings for Statistical Analysis

Once the analyst chooses a number of coders and the number of objects to be coded to create the training dataset, the next question is what is the optimal way to treat multiple codings of an object? Consider three possible methods. Assume one has N objects, each coded by at least K coders. First, one can simply take the modal (or mean) coding for each object. This clearly throws away information. In particular, this method retains our *best* coding of an object, but ignores the variance, or uncertainty, in our measure of the object. Second, one could 'stack' the data: treat each coding as an independent observation, and have $K \times N$

observations to train a classifier.[17,18] Third, we suggest that the analyst could make use of the variation across codings by treating the codings in the way one treats multiply imputed data, and making use of the machinery—and logic—of multiple imputation. Below we describe our proposed method for Classifier Training using Multiple Codings (CTMC).

As discussed above, we view each coding of an object as drawn from a distribution with a mean of the true value of the coding, and some variance. We want to take advantage of multiple codings by utilizing the fact that we 'know' more about an object if K coders say the object has value Y_i, than we know if only the modal coder says the object has value Y_i. And we also know more about the object if the mean of K of the object has value Y_i, with the variance *across the coders* for the object being low σ_i than we do if the mean of K coding of the object is Y_i, but the variance across the coders for the object is very high.

By creating K distinct datasets, estimating our model on each dataset, and then combining the K individual models into one classifier by averaging across estimated model parameters we are able to take advantage of this. If there is little variance about Y_i, then we effectively count each observation as contributing towards the estimate of the model parameters. However, if there is high variance about Y_i, then the observation, and codings about it, will contribute less to the final estimate of model parameters.[19]

We ran a simulation to test the three methods of treating multiple codings: taking the mode; stacking the individual codings (data expansion); and CTMC. We created a dependent variable y with a bernouli distribution about p, where $p = F(\mathbf{Xb} \ + \ \mathbf{e})$, and $\mathbf{X}$ is a feature matrix with N rows and 100 columns, with $x_{ij} \ = \ \mathcal{N}(0, 1)$, and $\mathbf{b}$ is a coefficient vector of

[17] This approach assumes that random assignment of articles to training and test sets happens at the article level, and not the coding level. This ensures that different codings for the same article are not in both training and test set, which could lead to overfitting.

[18] This would seem to violate an independence assumption in computing any standard errors of our estimates, but since we are not interested in the precision of the estimates, but only the fit of the classifier, this may not be an issue.

[19] One potential problem here is that some classifiers may set *different* model parameters to zero depending on the penalty function chosen. However, since we can of course average in zeros when combining estimates across datasets - this is a programming inconvenience but not a statistical issue.

[2, -2, 1, -1, 0, 0, 0 . . . , 0, 0, 0], of length 100, and $\mathbf{e}$ is random noise, with $\mathbf{e} = \mathcal{U}(-1, 1)$. We create K coders of quality level α, where α is the percentage of objects that a coder of quality α codes correctly. Thus a coder of quality level 100 has 0 variance, they code every object correctly.

We considered cases with 500, 1000, and 2000 objects to be coded, and with 3 and 5 coders per case, with coders of quality 0.50, 0.55, ..., 1.00. We then simulated 100 draws of all possible cases with combinations of these parameters, and estimated a regularized logistic classifier on each draw *using each of our 3 methods for multiple codings*: use the modal coding, stack the coding, or use the multiple imputation method. We computed the accuracy of the classifier for each possibility. We plot the results in Figure 1.

[**Figure 1 Here**]

In each case we can see that using the CTMC method produced higher accuracy; the CTMC line is always higher than the other lines. Not surprisingly as coder quality increased (i.e, each coder's variance decreased), the three methods generally converged given a large enough training dataset. We think the CTMC method is the best method to use for efficiently handling multiple codings.[20] As the sample size increases or coder quality increases the differences between the methods begin to disappear, but even with a sample of 2,000, regardless of the coder quality, CTMC outperforms the other methods of classification.

2 Evidence

In what follows we demonstrate the consequences associated with the different choices an analyst might make when confronted with the decisions detailed above. Our goal in the running example is to develop the best measure of tone of coverage of the U.S. national economy in the *New York Times* over the period 1947 to 2014. We begin by assessing the

[20] In analyses that follow in this paper we have not yet applied CTMC to our data, but while it would improve our machine learning classifiers, it would not change any inferences we draw throughout the paper.

differences in the size and nature of the corpora and resulting measures of tone created using two reasonable strategies for identifying the population of relevant articles from this universe of text (but limited to 1980-2011). The first corpus is that generated by Soroka et al. (hereafter *SSW*). It relies on a combination of Lexis-Nexis subject and sub-categories. The second corpus (hereafter *BBLMN*) is generated from an extensive list of keywords (section 2.1). In section 2.2 we consider the effects of the set of decisions analysts must make in creating the training dataset. We begin by assessing the effects of sentence versus article level coding by comparing out-of-sample accuracy of classifiers built from sentence and article level human coding (section 2.2.1). Then we evaluate the quality of coding by Penn State undergraduates versus CrowdFlower workers using the variance of the two groups of coders (section 2.2.2). Next we demonstrate the trade-offs between coding more objects (articles) versus adding more coders per object, again in terms of out-of-sample accuracy (section 2.2.3). In section 2.3 we end by comparing the out-of-sample accuracy of SentiStrength (Thelwall et al. 2010) and Lexicoder (Young & Soroka 2012) dictionaries, the Hopkins 9-Word method (Hopkins 2010), and 2 machine learning classifiers. In each section we offer suggestions to help analysts make these decisions in their own work.

Throughout, unless otherwise noted, the binary classifier was trained from coding produced using a 9-point ordinal scale (where 1 is very negative and 9 is very positive) collapsed such that 1-4=0, 6-9=1, and the midpoint (5) was omitted. The machine learning algorithm used to train the classifier uses logistic regression with an L2 penalty where the features are the 75,000 most frequent stemmed unigrams, bigrams, and trigrams appearing in at least 3 documents and no more than 80% of all documents (stopwords are included).[21] For the purpose of assessing out-of-sample accuracy we have two "ground truth" datasets. The first is a set of 442 articles coded by 10 CrowdFlower workers with high (70% or more) agreement (hereafter CF truth). The second such dataset includes 250 articles coded by an

[21] We compared the performance of a number of classifiers with regard to accuracy and precision in both out-of-sample and cross-validated samples before selecting logistic regression with an L2 penalty. See the appendix for details.

average of 5 undergraduate students that participated in a training session (hereafter UG truth). A summary of the various datasets is included in Table 1.

[Table 1 Here]

2.1 Comparing Strategies to Select the Corpus

A number of analysts have coded the tone of news coverage of the U.S. national economy. The universe of text defined in this body of work varies widely from headlines or front page stories in the *New York Times* (Blood & Phillips 1997, Wu et al. 2002, Fogarty 2005, Goidel & Langley 1995) to as many as 30 newspapers (Doms & Morin 2004). In some cases the full universe of text is coded while in others subject category and/or keyword searches have been conducted in an effort to produce a population of stories about the economy.[22] In a recently published article in *American Journal of Political Science*, Soroka, Stecula, and Wlezien (2015) estimated media tone about the economy for the period 1980 thru 2011. Their universe of media outlets consisted of the *Washington Post* and the *New York Times.* Their strategy for identifying articles about the U.S. economy relied on a combination of Lexis-Nexis subject categories and sub-categories. In contrast, our strategy relied exclusively on a keyword search. On the face of it, there is little reason to claim that one strategy will necessarily produce a better measure of tone. One might even expect the curated category search to produce a better corpus for measuring tone because it could have a higher proportion of relevant articles. However, as we argued above, an advantage of using the keyword search is that the method is reproducible to other publications and even other languages. Importantly for our purpose here, as we illustrate below the methods produce surprisingly distinct corpora of text and measures of tone of media coverage of the U.S. national economy. And despite expectations to the contrary, the proportion of articles deemed relevant is higher in the keyword search than in the category search.

[22] Blood and Philips (1997) define the universe as all headlines, and code them all.

Our strategy is to compare the corpus produced by the subject categories and sub-categories *SSW* identified with that produced using our keyword search.[23] We begin by comparing the total number of articles in each corpus in each year and the extent of the article overlap in each corpus. We then compare the monthly count of articles containing each of a set of economic terms in each corpus. Next we compare the proportion of relevant articles in each corpus. Finally, we compare the estimate of tone derived from each corpus using a) the dictionary Lexicoder (Young & Soroka 2012) and b) supervised machine learning.

The search listed by *SSW* captured articles that were indexed in at least one of the following Lexis-Nexis defined sub-categories of the subject "Economic Conditions": "Deflation", "Economic Decline", "Economic Depression","Economic Growth", "Economic Recovery", "Inflation", or "Recession". They also captured articles in the following Lexis-Nexis sub-categories of the subject "Economic Indicators": "Average Earnings", "Consumer Credit", "ConsumerPrices", "Consumer Spending", "Employment Rates", "Existing Home Sales", "Money Supply", "New Home Sales", "Productivity", "Retail Trade Figures", "Unemployment Rates", or "Wholesale Prices". The results of this search by *SSW* were also conditional on the articles having a relevance score of 85 or higher, as defined by Lexis-Nexis, for any one of the sub-categories listed above. Post-collection, *SSW* manually cleaned the set of documents by removing articles that were not focused solely on the domestic economy, were irrelevant, were shorter than 100 words long, or were "just long lists of reported economic figures and indicators," (Soroka, Stecula & Wlezien 2015, 461-462). However, due to concerns about reproducibility surrounding the manual cleaning of the corpus and potential changes to Lexis-Nexis since they downloaded the data, *SSW* generously provided us with their final data set of articles.[24]

[23] For parsimony's sake we only compare articles retrieved from the *New York Times* here.

[24] An example usage of this search is as follows: PUBLICATION (New York Times) AND DATE AFT(01/01/1995) AND DATE BEF(12/31/1995) AND (SUBJECT (deflation #85plus#) OR SUBJECT (economic decline #85plus#) OR SUBJECT (economic depression #85plus#) OR SUBJECT (economic growth #85plus#) OR SUBJECT (economic recovery #85plus#) OR SUBJECT (inflation #85plus#) OR SUBJECT (recession #85plus#) OR SUBJECT (Average Earnings #85plus#) OR SUBJECT (Consumer Credit #85plus#) OR SUBJECT (Consumer Prices #85plus#) OR SUBJECT (Consumer Spending #85plus#) OR SUBJECT (Employment Rates #85plus#) OR SUBJECT (Existing Home Sales #85plus#)

To generate our population of economic news stories, we downloaded all articles from the *New York Times* with **any** of the following terms in parenthesis (“employment”, “unemployment”, “inflation”, “consumer price index”, “GDP”, “gross domestic product”, “interest rates”, “household income”, “per capita income”, “stock market”, “federal reserve”, “consumer sentiment”, “recession”, “economic crisis”, “economic recovery”, “globalization”, “outsourcing”, “trade deficit”, “consumer spending”, “full employment”, “average wage”, “federal deficit”, “budget deficit”, “gas price”, “price of gas”, “deflation”, “existing home sales”, “new home sales”, “productivity”, “retail trade figures”, “wholesale prices”) **AND** “United States.”[25, 26] We applied a country filter to the set of articles downloaded from this search. We removed any article that mentions any country name, country capital, nationality or continent name (Schrodt 2011) in the headline or first 1000 characters of the articles and does NOT mention U.S., U.S.A. or United States in that same text fragment.

In Table 2 we show the number of articles in each corpus in each year from 1980 through 2011 (columns 1 and 2), as well as the overlap between them for each year (column 3).[27] Overall our corpus contained just under twice as many articles as did the *SSW* corpus (30,787 vs. 18,895). In general in years where our corpus contained relatively more articles, so, too, did the *SSW* corpus (the correlation between the annual counts is $\rho = .71$). But in some months our corpus contained over 3 times as many articles as the *SSW* corpus, while in others both corpora contained similar counts, and finally, in one year (2011), the *SSW*

OR SUBJECT (Money Supply #85plus#) OR SUBJECT (New Home Sales #85plus#) OR SUBJECT (Productivity #85plus#) OR SUBJECT(Retail Trade Figures #85plus#) OR SUBJECT (Unemployment Rates #85plus#) OR SUBJECT (Wholesale Prices #85plus#))

[25] In collecting this dataset, we obtained articles from two sources: the ProQuest Historical New York Times Archive and the ProQuest Newsstand Database. Articles in the first database span the 1947-2010 period and are only available in PDF format, and thus had to be converted to plain text using software. Articles for the 1980-2014 period are available in plain text through ProQuest Newsstand. We used machine learning techniques to match articles in both datasets and to delete duplicated articles, keeping the version available in full text through ProQuest Newsstand.

[26] We note that we could have generated the keywords using a(n) (un)supervised method of keyword generation or query expansion (King, Lam & Roberts 2016, Xu & Croft 1996, Rocchio 1971, Schütze & Pedersen 1994, Bai et al. 2005, Mitra, Singhal & Buckley 1998). In a topic less well defined than ‘the economy’, this is what we would likely do. However, in this case we felt that ‘the economy’ is well understood and we trusted our expertise.

[27] This range covers the entirety of *SSW*’s data.

corpus contained slightly more articles.

[**Table 2 Here**]

It is possible that the *SSW* search is essentially a subset of ours and that they simply have a more focused set of articles. In order to determine how distinct the two corpora are, we identified the overlap in articles appearing in each.[28] The third column of Table 2 presents the total number of articles in both corpora (common) while the remaining columns provide the number of articles unique to the BBLMN corpus (column 4) and the *SSW* corpus (column 5). Overall only 13.9% of our articles are included in the Soroka, et al. dataset; and only 22.7% of the Soroka, et al. articles were found in our dataset. This is surprisingly little article overlap between two corpora trying to measure the same thing. Perhaps a better way to illustrate this is to realize that we are trying to code reported media sentiment of the economy where 86.1% of our articles are *not* being used to code media sentiment by Soroka, et al.; and Soroka, et al. are trying to code reported media sentiment with a sample where 77.3% of their articles are not included in our corpus.

As an additional comparison between the two datasets, we present the monthly frequency of articles containing a set of terms central to economic performance included in each dataset in Figure 2: unemployment, inflation, stock market, and interest rates. The solid red line gives the number of articles in the *SSW* corpus in a given month that contain the term. The dashed blue line tracks the same counts in our dataset. The first thing to note (and consistent with Table 1) is that our keyword search consistently returns more articles

[28] In order to identify the overlap in the two datasets, we generated a set of potential matches between the datasets by searching for articles with similar headlines and that were published in the same year. (Article headline similarity was defined as having a maximum Levenshtein edit distance of 0.35.) This produced a set of *SSW* articles that were potential matches for each article in our data (we constrained the sets to have a maximum of ten potential matches). We then randomly sampled 25% all potential matches (2600 pairs). These sampled article pairs were subsequently coded as being true or false matches and then used as training data for an AdaBoost classification tree algorithm using the "caret" package in R. We were able to achieve a 10-fold cross-validated accuracy of $\sim 99.3\%$. Baseline accuracy over the entire training set was 58.7%. The resulting model was used to classify headlines as unique or matched for the remaining 75% of potential matches. Unique articles could be labeled as matches in multiple sets of article pairs. Pairs of articles containing non-unique article IDs were cleaned by hand. This form of duplication was exceedingly rare. In total, only 145 of 4,368 pairs of articles contained a non-unique article identifier.

with these terms, with the exception of articles containing the word unemployment, which appear with similar frequencies in each corpus. In general this is unsurprising given that each term was a keyword in our search while they needed to be included as a sub-category within the broader subject category of "Economic Conditions" (inflation) or "Economic Indicators" (unemployment rates) or were not included as either a subject or sub-category (interest rates and stock market). The correlations in these term frequencies are reasonably high, ranging from a low of 0.69 (stock market) to a high of 0.91 (unemployment). But across the full set of terms correlations are often lower: bond (0.64), federal reserve (0.65), employment (0.66), household income (0.59), and average wage (0.47). Thus, in general, in months where our search returns more articles containing a given term, so too does the *SSW* search. But these correlations are not perfect, suggesting the two corpora are different in ways that may be important.

[**Figure 2 Here**]

It appears that using the combined subject category and sub-category search narrowed the set of articles in the *SSW* population considerably. If we feel any search strategy should retrieve articles containing 'unemployment', 'stock market', 'inflation', and 'interest rates,' then the *SSW* search is too narrow, potentially omitting relevant stories. But perhaps our search is too broad and the use of subject (and sub-) categories eliminates articles that provide no information about the state of the U.S. national economy, in spite of containing these keywords. One way to assess these possibilities is to compare the proportion of articles in each corpus that provide information about the state of the economy and those that do not. To perform this analysis we had three CrowdFlower coders code the relevance of 1000 randomly selected articles a) unique to the *SSW* corpus, b) unique to our corpus, and c) in both corpora. We present the results in Table 3.[29]

[29] All three coders coded each article producing 9000 total codings. We computed the proportion relevant by aggregating to the article level. Each coder was assigned a weight based on her overall performance before computing the proportion of articles deemed relevant. If two out of three (weighted) coders concluded an article was relevant, the aggregate response is coded as "relevant". The coding-level proportions were qualitatively equivalent and are presented in the appendix.

[Table 3 Here]

The proportion of relevant articles is given in the top row of Table 3. Unsurprisingly we find that in articles in both the *SSW* corpus and our corpus the proportion of relevant articles was the highest (column 1); just under half of the articles were coded as relevant (0.44). For articles in our corpus but not *SSW's* (column 2) the proportion is very similar (0.42). The proportion of relevant articles in the *SSW* corpus but not ours (column 3) drops to 0.37.[30] Overall the results suggest both search strategies are too broad, picking up a large number of irrelevant articles. The more narrow *SSW* search did not produce a higher proportion of relevant articles, suggesting that the Lexis-Nexis subject categories do not provide any reassurance that an article will provide "information about the state of the economy". We note that, ceteris paribus, if two searches produce corpora containing identical proportion of relevant articles (i.e., articles that are actually elements of the population set) then the larger search is to be preferred: it is going to generate a larger sample of text from which we will ultimately estimate the population parameters (e.g., tone) of interest.

These comparisons suggest both corpora contain large portions of irrelevant articles and that each corpus is highly distinct. But do these differences matter? The large proportion of irrelevant articles suggest measures of tone produced from each corpus will contain measurement error. The highly unique content of each corpus suggests the potential for bias in both measures of tone. But given that we do not know the true tone as captured by a corpus that includes all relevant articles and excludes all irrelevant articles, i.e., in the population of articles on the U.S. national economy, we cannot address these concerns directly. We can, though, compare the measures of tone produced a) by the application of Lexicoder (Young & Soroka 2012) to each corpus and b) the application of supervised machine learning to each corpus to determine how much the differences in corpus affect the estimated measures of tone.[31] Applying Lexicoder to both corpora, we find a correlation of

[30] Coders were unsure of the relevance of so few articles that the proportions coders deemed not relevant is simply 1- proportion relevant.

[31] Lexicoder sentiment scores for documents are calculated by taking the number of positive minus the

0.48 between the two monthly series of tone produced over the overlapping time period. Our supervised algorithm produces monthly series of tone from each of the two corpora that are correlated with a value of 0.59 over the same time span (a substantially higher correlation than between the two series produced by Lexicoder). These correlations are surprisingly low for two measures purported to measure the same concept.

We have learned that different searches can produce different datasets that result in different estimates of tone but we have no way to know the truth to say which search is better. The problem is likely to be worse in less well-defined topics. It is a reminder that as we move to machine-learning to code sentiment we are forced to confront one of the oldest adages of coding: Garbage-In, Garbage-Out (GIGO). Failing to accurately consider the quality (i.e., relevance) of the data we will code is likely to have a large impact on the outcome. We recommend the analyst conduct extensive pretesting, including coding the sample of text for relevance.[32] There is a simple algorithm the analyst can follow: a) do a narrow key-word search; b) do a broad key-word search; c) code a sample of each corpus for relevance. IF (b) returns way more objects than (a), AND relevance does not go down, then using (a) would leave out many relevant articles *and* potentially bias the measure produced.

We have seen in the case at hand that a keyword-based search produces a larger corpus of articles with a higher rate of relevancy than a topic-supplied-by-media-maker search does. We do not know the true measure of tone of the entire population of media coverage in the *New York times* over the period 1980-2011. However, we again default to our notion that we are inherently sampling here—and everything we know about inferring population parameters from samples applies. All else being equal, we will get a better estimate of the

number of negative terms over the total number of terms (Eshbaugh-Soha 2010, Soroka, Stecula & Wlezien 2015). The machine learning algorithm uses logistic regression with an L2 penalty, where the features are the 75,000 most frequent stemmed unigrams, bigrams, and trigrams appearing in at least 3 documents and no more than 80% of all documents (stopwords are included). It is trained on the crowd-coded article level corpus and then applied to both corpora separately. The monthly tone estimates for both measures are the simple averages across the articles in a given month.

[32] Keyword expansion is unlikely to be a solution to this problem – but may be wise – as it does not inform the analyst whether the search is too broad or too narrow, i.e., when do we stop expanding?

population parameter with a larger sample. And the keyword search is producing a larger sample of relevant articles (i.e., articles that truly belong to the population we are trying to measure). And it is not doing so at a cost of higher noise: the proportion of irrelevant articles in the keyword dataset is lower than the proportion of irrelevant articles in the SSW dataset.

2.2 Making Decisions about the Training Dataset

Below we discuss issues involved in creating the training dataset to be used for supervised machine learning. When the analyst starts, the tone of articles is unobserved. Humans will be used to code some sub-sample of articles for tone. At that point, the analyst has a set of articles where the dependent variable is observed, and the analyst can attempt to estimate a model based on the features of the articles (the text) that would predict the dependent variable. However, the analyst is of course observing the dependent variable with error. And as we have discussed above, that error will be a function of the accuracy of the coders, and the number of coders used per article. There also may be error from other sources. The analyst is using a survey question to elucidate tone, thus the nature of the survey question also matters. And, the unit of analysis problem must be dealt with: humans could coded anything from individual words to entire articles.

2.2.1 Comparing Units of Analysis

In section 1.4.1 we considered the trade-off between sentence and article level coding: sentence level coding provides more precise information for the classifier but article level coding is less costly on a per article basis. We evaluate the effect of unit of analysis by first subjecting a set of 2000 articles to human coding at the sentence level and the article level.[33, 34] Then we trained a classifier on both the sentence and article level training datasets. Next, the parameters estimated by each classifier were used to predict the tone of articles. Finally, we compared out-of-sample accuracy of article classification based on each level of analysis in CF truth.

It appears that the trade-off between specificity of coding and increased cost has, in this case, surprisingly little consequence. The accuracy of the data coded at the sentence and article level are 0.700 and 0.693, respectively. This exercise suggests that decreased specificity associated with a more coarse, article level coding scheme *need not* diminish accuracy and thus that the cost savings associated with coding articles *may* tip the scale in their favor. It is unclear how this conclusion generalizes.

We recommend the analyst follow these same steps using a small test-batch to determine whether the additional information contained in the sentence level coding is worth the cost in her application. As an alternative, the analyst might first code a set of articles by sentence. To the extent that the sentence codings within an article are consistently positive (or neutral) or negative (or neutral), little information is likely to be lost by coding articles. If the distribution of positive and negative sentences across the articles is mixed, sentence level coding may be necessary to produce a measure of tone with the highest degree of accuracy.

[33] We did not actually provide coders with entire articles—but with approximately the first five sentences of each article. Text was produced via OCR processing on pdf files and we adopted an algorithm to approximate the first five sentences of text.

[34] The coding at the sentence and article level was based on a 9-point scale (from 1, very negative, to 9, very positive) where we recode to 1-4 to negative (0), 6-9 to positive (1), and exclude the rest for purposes of training the classifier. The machine learning algorithm uses logistic regression with an L2 penalty, where the features are the 75,000 most frequent stemmed unigrams, bigrams, and trigrams appearing in at least 3 documents and no more than 80% of all documents (stopwords are included).

2.2.2 Choice of Coders

Who should the analyst have code the data in the training dataset? In this section we compare the relative quality of (Penn State) undergraduate coders and CrowdFlower coders. In section 1.4.2 we argued that if, as we assume, each coder provides an unbiased estimate of the truth, the best metric for determining coder quality is the average object-specific variance of coders. We suggested undergraduate coders, because they are trained as a group, will have lower variance than crowd workers. To test this expectation, we examine the inter-coder reliability of both groups using our measure of average intra-object variance.[35] We also present a traditional measure of coder quality (average pairwise inter-coder agreement, APIA) as a point of comparison.

Table 4 reports both measures of inter-coder reliability for undergraduates and CrowdFlower workers.[36] We do this for: 1) undergraduates and CrowdFlower workers for coding on a common dataset (420 sentences); 2) undergraduates over the entire set of 1051 sentences coded as relevant by undergraduates; and 3) over the entire set of 1788 sentences coded as relevant by CrowdFlower workers. In the set of sentences coded by both undergraduates and CrowdFlower workers, we measure the variance of codings about each object and then average the object-specific variance as coded by each group.[37] These measures are directly comparable. To get the best measure we can of undergraduate variance and CrowdFlower coding variance, we repeat this process for the entire set of sentences coded by undergraduates, and the entire set of sentences coded by CrowdFlower workers.[38]

[Table 4 Here]

[35] More specifically, we compute the variance of the codings for each sentence, then take the average of the variances. This takes into account the varying number of coders per sentence.

[36] Sentences were coded by each group using a 5-category scheme (negative, mixed, neutral, not sure, positive). Responses were recoded to -1 (negative), 0 (mixed, neutral, not sure), and +1 (positive) before computing the variance.

[37] The number of coders for each sentence varied from between 2 and 10 undergraduates and 2 and 6 CrowdFlower coders. The average number of undergraduate coders for each sentence was approximately 3.1 while for CrowdFlower coders it was 2.4.

[38] These two estimates are not perfectly comparable as they are computed over different datasets and in particular because the sentences are not random draws of all sentences but rather 5 come from each article.

Our results are consistent with our expectations: undergraduate coders exhibit lower variance codings and higher average pairwise inter-coder agreement (APIA) than crowd workers. In the set of sentences coded by both groups, variance of crowd worker codings was 0.53 while that for undergraduates was almost half that (0.28). (See column 1, rows 3 and 4.) For the population of sentences coded by each group the variances were similar: 0.57 for crowd workers and 0.30 for undergraduates. (See column 1, rows 1 and 2.) The average pairwise agreement is also higher for the undergraduate coders than CrowdFlower coders (column 2). For the overlapping sample of sentences, APIA for undergraduate coders is 0.84 while that for crowd workers is 0.72. The comparisons are similar for the full set of sentences coded by each group (0.82 for undergraduates and 0.74 for crowd workers).[39]

With this information in hand and still assuming coders are unbiased, an analyst can determine the number of CrowdFlower coders (J^*) needed to get an average variance similar to the codings of undergraduates (J_1) for a fixed number of objects coded by all J_1 coders. We can approximate J^* as $\frac{0.57}{0.30} = 1.9$, which implies we need nearly twice as many CrowdFlower coders per sentence as undergraduate coders to get the same variance. Given an estimate of the relative cost of coding for each type of coder the analyst can make a determination of the most cost effective way to proceed. In our example, if the relative cost of CrowdFlower coders is about half (or less) that of undergraduate coders, the analyst would choose the CrowdFlower coders.

Ultimately budget—time and money—constraints will drive this decision. The analyst may be best served by coding a small set of objects (ideally by the same number of coders) in each potential coding pool and comparing the ratio of the *per coder* variance of the codings in each group with the ratio of the costs in each group. If the per coder variance ratio is greater than the cost ratio, the analyst should use larger numbers of the higher variance

[39] Ideally we would a) have each set of coders classify the same set of articles, b) train a classifier on each set; c) use those two classifiers to code articles in a dataset where truth is known; and d) compare out-of-sample accuracy. However, we did not have enough articles coded at the article level by undergraduate students to do so.

coders. This calculation does not tell us how many coders to use nor give us any purchase on the number of objects to code in the creation of the training dataset. We turn to these questions next.

2.2.3 How Many Coders? How Many Objects?

In preparing a training dataset, a critical choice is how many codings of each object to obtain. Assuming a fixed cost per coding of each article, the analyst could code N articles by 1 coder each, N/2 articles by 2 coders each, N/3 articles by 3 coders each, etc.[40] The best decision is a function of inter-coder reliability, or, as described above, variance in responses across coders. If all coders always agreed, then there is no point in getting more than 1 coding per article: the additional coding provides no new information. If coder variance is low there is less uncertainty about evaluations and the return to additional codings per article is low. If coder variance is high there is more uncertainty and the return to additional codings per article is relatively high. We test the return to additional codings in the case of coding economic sentiment by training a binary classifier on a training dataset of 4400 articles from the *New York Times* randomly sampled from the years 1947-2014. The data was coded by from 1 to 10 CrowdFlower coders using the 1-9 ordinal scale. The classifier was trained by collapsing the original coding such that 1-4=0, 6-9=1, and omitting the midpoint (5). To illustrate the real world trade-offs one makes between choosing between more coders and more articles, we did this for varying sized subsets of the full set of articles: from $1/10^{th}$ of the set to the entire set of 4400 articles. For each subset, we trained with 1 coder, 2 coders, 3 coders, ..., 10 coders.

We then examined the accuracy of the classifier produced by each combination of the number of articles and number of CrowdFlower coders, with classification of an article being deemed "accurate" if it matched coding in UG truth.[41] Figure 3 plots the accuracy for each

[40] Note that this changes if one is using undergraduates and has to train each coder—then the math is a bit different, and time constraints and labor constraints come into play.

[41] We cannot use CF truth to compute out-of-sample accuracy because this data served to train the

of the 100 combinations of number of articles coded and number of coders. We find that both adding coders and increasing the size of the dataset improves the accuracy of the estimator.

In the extreme, even 10 coders cannot overcome the lower accuracy rates associated with a small sample size (440). Likewise, increasing the sample size to 4400 while relying on one coder does not purchase an increase in accuracy rates. But we don't need either 4400 articles or 10 coders to achieve the highest accuracy rates. The biggest gains in accuracy, which in this case are quite small, happen when we add more coders to moderate-sized samples (around 2000).

However, it appears that after adding more than 3 coders, the gains in accuracy are less pronounced than equivalent increases in sample size. The magnitude of the tradeoff here is conditioned by the variance of the CrowdFlower coders, which was 0.716 on the scale used to train the classifier, 1-4=0, 6-9=1, 5 omitted. With higher variance coders, presumably the marginal utility of additional coders would be greater than that associated with coding additional objects, while with lower variance coders, the marginal gains associated with more objects coded would be greater than that associated with more coders.

[Figure 3 Here]

2.3 Machine Learning vs. Dictionary Methods

In this section we compare the performance of a set of supervised machine learning methods and dictionary methods. We begin with a short description of the dictionaries and the classifiers used in our comparisons. We then compare the out-of-sample accuracy (vis a vis CF truth and UG truth) of our best performing classifiers with that based on the application the Sentistrength (Thelwall et al. 2010) and Lexicoder (Young & Soroka 2012) dictionaries, as well as Hopkin's (2010) 9-Word Method.

classifier.

2.3.1 The Dictionaries

We consider two sentiment dictionaries frequently used in previous studies: a general sentiment dictionary, SentiStrength (Thelwall et al. 2010), and a sentiment dictionary applied specifically to political text, Lexicoder (Young & Soroka 2012). In brief, SentiStrength assigns scores to words and short phrases appearing in a given text (that are also included in its dictionary), with scores ranging from -5 (most negative) to +5 (most positive). To assign a sentiment score to the document, we subtract the negative scores from the positive scores for each article. In contrast, Lexicoder assigns every n-gram in a given text a binary indicator if that n-gram is in its dictionary, with markers so that the algorithm can count the number of zeros (negative words) and the number of ones (positive words). Sentiment scores for documents are then calculated by taking the number of positive minus the number of negative terms over the total number of terms. We also compare these two approaches to the relatively simple method proposed by Hopkins (2010), which consists of counting the number of articles per month mentioning nine economic words (inflat, recess, unempl, slump, layoff, jobless, invest, grow, growth).

2.3.2 The Classifiers

Selecting the optimal classifier to compare to the dictionaries requires a number of decisions that are beyond the scope of this paper (but see Raschka (2015), James (2013), Hastie (2009), Caruana (2006)). In addition to the decisions addressed in our paper, the analyst must make decisions about how to preprocess the text that include the following: whether to stem the text (truncate words to their base), how to select and handle stop words (commonly used words that do not contain relevant information), and the nature and number of features (n-grams) of the text to include. Here we summarize our decisions and findings. (We outline these issues and provide details of the estimation in the appendix.)

The training data for the classifier was generated from 4400 articles in the *New York*

Times randomly sampled from the years 1947 to 2014. Each article was coded at the article level by up to three CrowdFlower coders using the 9-point coding instrument. For purposes of training the classifiers we recode 1-4=0 and 6-9=1, omitting articles coded as "not relevant" or at the midpoint (5). The optimal classifier was selected from a set of single-level classifiers including logistic regression (with L2 penalty), Lasso, ElasticNet, SVM, Random Forest, and AdaBoost.[42] Based on accuracy and precision evaluated using UG truth and CF truth, we selected the regularized logistic regression with L2 penalty with up to 75,000 n-grams appearing in at least 3 documents and no more than 80% of all documents, using stopwords and stemming. We also chose to take advantage of the specific application and code co-occurrences of certain combinations of terms as features of articles. For instance, we know that if 'up' or 'rising' occurs with 'unemployment', that is an indicator of negative tone—whereas the presence of any of those words on their own could indicate either negative or positive tone.

We report the n-grams that are most predictive of positive and negative tone in Table 5. Prima facia inspection indicates that we are capturing sentiment when we apply our classifier to this dataset. The top predictive negative n-grams (stemmed) include "declin", "recess", "cost", unemploy", "slump", "deficit", "plung", "fear", "worst", and "layoff", words closely associated with a poor economy. Similarly the list of top predictive positive n-grams begins with a set of words we associate with a strong or improving economy: "gain", "strong", "rise", "growth", "advanc", "recoveri". The lists also include less obvious n-grams, namely "washington", "american", "school", "day" (negative n-grams) and "new york", "januari", and "person" (positive n-grams). But if these words frequently co-occur in articles with other highly predictive n-grams, these too, will contribute to accurate predictions of article tone.

[Table 5 Here]

[42] One could attempt to simultaneously model relevance and tone, or to simultaneously model topics and then assign tone within topics—allowing the impact of words to vary by topic. Those are considerations for future work.

2.3.3 Results of Supervised Machine Learning versus Dictionary

We are now ready to compare the performance of the logistic regression classifier with L2 penalty using the baseline decisions described above and the classifier based on co-occurrences of directional terms with words describing the economy to that of Lexicoder, Sentistrength and the Hopkins 9-Word Method. We evaluate performance with reference to articles for which human coders are in substantial agreement about the tone of the content. Specifically, using each of the 5 approaches, we predict the tone of of articles in our ground truth datasets—UG Truth ($N = 250$) and CF Truth ($N = 404$)—and compare the percentage of articles for which each approach correctly predict the direction of tone.

Each row of Figure 4 reports the accuracy of each classifier method on the two truth datasets. In each figure we include a dashed line to represent the baseline level of accuracy (percentage of articles in the modal category). Machine learning approaches outperform dictionary methods, which do not improve over a random benchmark (predicting the modal category for all articles). More specifically, we find our baseline classifier has a 74.0% accuracy rate in UG Truth. In other words, the baseline classifier correctly predicted coding by Penn State undergraduates in 74% of articles they coded. Adding directional terms improves accuracy slightly (74.4%). This compares to 57.2% for SentiStrength and 57.6% using LexiCoder. The Hopkins 9-Word method has an accuracy of less than half that of the machine learning classifiers (35.6%). Similar accuracy is obtained in CF Truth, with only the 9-Word method improving (39.9%), although still performing much worse than the alternatives. Only machine learning classifiers achieved an accuracy rate over a random benchmark.

[Figure 4]

These results provide very strong evidence for the superiority of supervised machine learning over dictionary methods for measuring the quantity of interest. Our goal is to produce a measure of the tone of economic coverage as conveyed to readers by the media.

Whatever the arguments are in favor of dictionary methods, if the dictionary produced measures of tone do *not* correspond to what human readers would perceive - then the dictionaries have failed and we are better served by other methods. While the machine learning methods we report here fall short of the level of accuracy we might want, they do perform substantially better than the dictionary methods. And aside from poor performance as revealed in Figure 4, dictionaries also suffer from a lack of observable measurement validity. In the absence of testing the dictionaries as we did here, the analyst would not know how badly they are performing at the core task of measuring tone as readers perceive it. We think that this argues strongly for preferring supervised machine learning over dictionary methods.

3 Conclusion

Producing a scalar measure of some quantity from text has become easy. The analysis of text for meaningful social science research remains hard. For those engaged in the enterprise, the first tasks are to identify a population of relevant texts and decide a classification approach. If using supervised machine learning to generate a measure of tone, one must make a number of decisions about the training dataset. What is the optimal unit of analysis to code, who will code the data, how many coders and how many objects to code, and how to make use of multiple codings. The consequences of all of these decisions have not been well understood and, as we demonstrated, are not benign. Our goal was to lay out the costs and benefits of a number of different strategies to select answers to these questions and to offer some advice on how to navigate them. We summarize our recommendations below.

Identifying the Population Corpus. While text is widely electronically available, identifying the universe of text and sampling from it to create a population of "relevant" texts requires the analyst to make a number of decisions and be aware of the limits of text archives. Do we use subject categories, keywords, or something else? After downloading texts, (how) do we filter the data to define the corpus? As we showed, these decisions are not without consequences. Furthermore, as social scientists, we place high value on the ability to reproduce the corpus and to define a strategy that is transportable across media sources, including those in other countries. Analysts must also be aware of decisions by archivers that are outside the control of the analyst and that affect our ability to approximate the population of relevant texts as well as maintain reproducibility of the corpus. The analyst needs to answer several questions. How does the universe of text electronically available compare to that in the print source? What's added and subtracted from the databases we draw from over time?

Given the twin goals of including all (and only) relevant texts in the corpus used for analysis, and producing an algorithm that is transportable and reproduces a given corpus, we strongly recommend analysts avoid proprietary—and thus nontransparent, nontransferable, and time varying—subject classifications to generate the corpus of text. Instead we recommend keyword searches. They are transparent, transferable, and can be applied consistently over time. Further, the breadth of the search is in the control of the analyst. We further recommend the analyst begin by applying both a relatively narrow keyword search and a broader search and code a sample of each set of texts for relevance. If the broader search returns more objects and relevance does not go down, the broader search should be adopted as using the narrow search would exclude relevant articles and potentially bias the resulting measure of tone. If the narrow search produces a larger proportion of relevant texts, then the analyst must consider the tradeoff between the smaller set that can be more easily coded, but *may* cause bias by ignoring *non-random* articles, and the larger set that is harder to code and will include more noise. The strategy of keyword searches with tests for relevance allows the analyst to optimize the search criteria in designing the population corpus *prior* to the stage of producing the final training set and estimating the parameters of the classifier.

Classifier Method: Should the analyst use dictionary methods or supervised machine learning? We strongly recommend against the use of dictionaries—even those designed to measure economic sentiment—and recommend a supervised machine learning method. Dictionaries define relevant features of the text and the weight attached to them a priori. Generally all features have the same weight. Surely these are untenable assumptions. And after the analyst has applied the dictionary, formal methods of assessment (absent any human coding) are limited to tests of convergent validity, a strategy that mixes evaluation of the measure with hypothesis tests about the relationship between the measure and other concepts. In contrast, SML methods, while unequivocally more complicated, offer several advantages. Of particular importance, the relevant features of the text and their weights are estimated directly. Further, because the analyst possesses a set of human coded text, assessments of the classifier can be made relative to human understanding of the text. Finally, as demonstrated here, SML produced a measure of the tone of text that substantially outperformed a number of dictionary approaches *designed specifically to measure the tone of economic news coverage* as assessed relative to human coding. We see no reason to expect that this conclusion does not generalize.

Decisions about the training dataset: How should the analyst select a unit of analysis? A set of coders? The number of coders per object and number of objects to code? Our advice on these questions is more circumspect. Absent test analyses in the analyst's particular application, we offer no universal answers to these questions. When deciding the unit of analysis, intuition suggests sentence level coding is more precise, but in any given application article level coding may be sufficient (and cheaper). We recommend analysts code a subset of articles at the sentence level and analyze the distribution of features of the text across sentences. If sentences within articles typically have the same valence, coding articles is likely sufficient. However if sentences have features with the same valence while articles do not, sentence level coding may be necessary to achieve optimal levels of precision.

As we explained above, the question of whom to code the data, how many coders to employ per object, and how many objects to code cannot be considered in isolation. Why? We want to minimize measurement error in our best prediction of the tone of an object. If coders are unbiased then measurement error will be smaller with low variance coders, a larger number of coders and a larger number of objects coded, all else equal. But these factors interact. Given low variance coders, the number of coders and objects coded need not be as great as with higher variance coders. At the same time lower variance coders, such as trained undergraduates, are likely to be more costly—both in dollars and in time per coding. Thus we recommend analysts assess coder quality as estimated by the average item variance per coder in any potential coding pools and determine whether it is more cost-efficient to code with more lower quality coders or fewer higher quality coders.

The twin questions of number of coders and number of objects coded will depend on the quality (variance) of coders. The analyst could compare the variance of different potential groups of coders, produce a measure of tone with each group of coders and given a fixed number of coders and objects coded, and assess the relative accuracy gains associated with each of these factors. This would enable her to assess the various trade-offs directly. But this is a burdensome task. We plan to conduct simulations to try to provide more useful answers to these questions. Based on our work so far, what do we recommend the analyst do? Our limited evidence suggests a minimum of 3 coders and about 1,000 relevant objects are necessary to achieve higher out-of-sample accuracy rates without substantially more investment in either number of coders or codings. The diminishing returns to more coders per article will set in sooner with lower variance coders. In the end, the test of all elements of the coding scheme (type of coders, number of coders, number of objects coded) is based on the quality of the classifier produced, *not* the inter-coder reliability between coders.

A Parting Word. The opportunities afforded by vast electronic text archives and machine classification of text for the measurement of a number of concepts, including tone, are in a real sense unlimited. In a rush to take advantage of the opportunities, we have to date overlooked some important questions and under appreciated the consequences of some decisions. It is extremely easy for an analyst to acquire a corpus of text, apply some coding scheme, and produce a measure of the concept of interest. But this is no guarantee the resulting measure is adequate to the task. Careful consideration must be paid to the questions covered above (and others). Our most basic advice is: classify by machine, but verify by human. No machine generated measure of a latent characteristic of text should be trusted if it has not been verified by comparing it to human perception of the text.

References

Bai, Jing, Dawei Song, Peter Bruza, Jian-Yun Nie & Guihong Cao. 2005. Query expansion using term relationships in language models for information retrieval. In *Proceedings of the 14th ACM international conference on Information and knowledge management*. ACM pp. 688–695.

Blood, Deborah J & Peter CB Phillips. 1997. “Economic headline news on the agenda: New approaches to understanding causes and effects.” *Communication and democracy: Exploring the intellectual frontiers in agenda-setting theory* pp. 97–113.

Caruana, Rich & Alexandru Niculescu-Mizil. 2006. An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd international conference on Machine learning*. ACM pp. 161–168.

De Boef, Suzanna & Paul M Kellstedt. 2004. “The political (and economic) origins of consumer confidence.” *American Journal of Political Science* 48(4):633–649.

Doms, Mark E & Norman J Morin. 2004. “Consumer sentiment, the economy, and the news media.” *FRB of San Francisco Working Paper* (2004-09).

Eshbaugh-Soha, Matthew. 2010. “The tone of local presidential news coverage.” *Political Communication* 27(2):121–140.

Fan, David, David Geddes & Felix Flory. 2013. “The Toyota recall crisis: Media impact on Toyota’s corporate brand reputation.” *Corporate Reputation Review* 16(2):99–117.

Fogarty, Brian J. 2005. “Determining economic news coverage.” *International Journal of Public Opinion Research* 17(2):149–172.

Goidel, Kirby, Stephen Procopio, Dek Terrell & H Denis Wu. 2010. “Sources of economic news and economic expectations.” *American Politics Research* .

Goidel, Robert K & Ronald E Langley. 1995. “Media coverage of the economy and aggregate economic evaluations: Uncovering evidence of indirect media effects.” *Political Research Quarterly* 48(2):313–328.

Grimmer, Justin & Brandon M Stewart. 2013. “Text as data: The promise and pitfalls of automatic content analysis methods for political texts.” *Political Analysis* 21(3):267–297.

Grimmer, Justin, Gary King & Chiara Superti. 2015. “The Unreliability of Measures of Intercoder Reliability, and What to do About it.” Working paper.

Grimmer, Justin, Solomon Messing & Sean J Westwood. 2012. “How words and money cultivate a personal vote: The effect of legislator credit claiming on constituent credit allocation.” *American Political Science Review* 106(04):703–719.

Hastie, Trevor, Robert Tibshirani & Jerome Friedman. 2009. Unsupervised learning. In *The elements of statistical learning*. Springer pp. 485–585.

Hillard, Dustin, Stephen Purpura & John Wilkerson. 2008. "Computer-assisted topic classification for mixed-methods social science research." *Journal of Information Technology & Politics* 4(4):31–46.

Hopkins, Dan. 2010. "Does Newspaper Coverage Influence or Reflect Public Perceptions of the Economy?" *APSA conference paper* .

James, Gareth, Daniela Witten, Trevor Hastie & Robert Tibshirani. 2013. *An introduction to statistical learning*. Vol. 6 Springer.

Jurka, Timothy P, Loren Collingwood, Amber E Boydstun, Emiliano Grossman & Wouter van Atteveldt. 2013. "RTextTools: A supervised learning package for text classification." *The R Journal* 5(1):6–12.

King, Gary, Patrick Lam & Margaret Roberts. 2016. "Computer-Assisted Keyword and Document Set Discovery from Unstructured Text." Working Paper.

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

Loughran, Tim & Bill McDonald. 2011. "When is a liability not a liability? Textual analysis, dictionaries, and 10-Ks." *The Journal of Finance* 66(1):35–65.

Mitra, Mandar, Amit Singhal & Chris Buckley. 1998. Improving automatic query expansion. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*. ACM pp. 206–214.

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

Raschka, Sebastian. 2015. *Python Machine Learning*. Packt Publishing Ltd.

Rocchio, Joseph John. 1971. "Relevance feedback in information retrieval.".

Schrodt, Phil. 2011. *Country Infro, 111216.txt*.
URL: *https://github.com/openeventdata/CountryInfo*

Schütze, Hinrich & Jan O Pedersen. 1994. A cooccurrence-based thesaurus and two applications to information retrieval. In *Intelligent Multimedia Information Retrieval Systems and Management-Volume 1*. LE CENTRE DE HAUTES ETUDES INTERNATIONALES D'INFORMATIQUE DOCUMENTAIRE pp. 266–274.

Soroka, Stuart N, Dominik A Stecula & Christopher Wlezien. 2015. "It's (Change in) the (Future) Economy, Stupid: Economic Indicators, the Media, and Public Opinion." *American Journal of Political Science* 59(2):457–474.

Tetlock, Paul C. 2007. "Giving Content to Investor Sentiment: The Role of Media in the Stock Market." *The Journal of Finance* 62(3):1139–1168.
URL: *http://dx.doi.org/10.1111/j.1540-6261.2007.01232.x*

Thelwall, Mike, Kevan Buckley, Georgios Paltoglou, Di Cai & Arvid Kappas. 2010. "Sentiment strength detection in short informal text." *Journal of the American Society for Information Science and Technology* 61(12):2544–2558.

Wu, H Denis, Robert L Stevenson, Hsiao-Chi Chen & Z Nuray Güner. 2002. "The Conditioned Impact of Recession News: A Time-Series Analysis of Economic Communication in the United States, 1987–1996." *International Journal of Public Opinion Research* 14(1):19–36.

Xu, Jinxi & W Bruce Croft. 1996. Query expansion using local and global document analysis. In *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval.* ACM pp. 4–11.

Young, Lori & Stuart Soroka. 2012. "Affective news: The automated coding of sentiment in political texts." *Political Communication* 29(2):205–231.

Table 1: **Details for Data Used in the Analysis**

Analysis (Section)	Sample Size[a] (Unit of Analysis)	Number & Type of Coder	Coding Scale	Truth Dataset
Unit of Analysis	10,000[b](S)	3 CF	9-point scale[c]	CF Truth
(Section 2.2.1)	2,000(A)	3 CF		
Coder Quality	420(S)	2-14 (Mean 3.1) UG	5-category[d]	NA
(Section 2.2.2)	420(S)	2-6 (Mean 2.4) CF	5-category	NA
	1051(S)	2-14 (Mean 3.1) UG	5-category	NA
	1788(S)	2-6 (Mean 2.4) CF	5-category	NA
# Coders, # Articles	4400(A)	1-10 (Mean 3.2[g]) CF	9-point scale	UG Truth
(Section 2.2.3)				
Dictionary v SML	4400(A)	1-3 (Mean ?) CF	9-point scale	UG Truth
(Section 2.3)				CF Truth
UG Truth[e]	250(A)	1-8 UG	5-category	
CF Truth[f]	442(A)	10 CF	9-point scale	

Note: All articles in the training data were randomly selected over the period 1947-2014.
The machine learning algorithm used to train the classifier uses logistic regression with an L2 penalty, where the features are the 75,000 most frequent stemmed unigrams, bigrams, and trigrams appearing in at least 3 documents and no more than 80% of all documents (stopwords are included). All CrowdFlower coders were located in the U.S. The two 4400 article datasets comprise distinct datasets.

[a] Reported sample size includes objects coded as irrelevant by all coders. Effective sample size ranged from was typically about 40% smaller.

[b] Five sentences were coded in each of the articles; the 2,000 articles were the same articles used to code 10,000 sentences.

[c] The 9-point scale ranged from 1 (very negative) to 9 (very positive). Categories were collapsed such that 1-4=0 (negative), 6-9=1 (positive) with the midpoint (5) dropped for purposes of training the classifier.

[d] The 5-category coding scheme allowed coders to label a sentence as negative, mixed, neutral, not sure, positive. Responses were recoded to -1 (negative), 0 (mixed, neutral, not sure), and +1 (positive) before computing the variance.

[e] The "truth" in UG Truth is based on students with high agreement.

[f] The "truth" in CF Truth is based on 70% or higher agreement among coders.

[g] The mean number of coders reflects the average number of coders per object who coded the tone of the object (excluding objects coded as irrelevant)

Table 2: Comparing the *SSW* Corpus with Our Corpus: Total, Unique, and Overlapping (Common Corpus) Article Counts

	BBLMN	*SSW*	Common	Unique	Unique
Year	Corpus	Corpus	Corpus	BBLMN	*SSW*
1980	1767	516	73	1694	443
1981	1545	945	133	1412	812
1982	1960	1361	344	1616	1017
1983	1618	840	206	1412	634
1984	1304	629	112	1192	517
1985	1103	481	85	1018	396
1986	1020	444	84	936	360
1987	1340	552	93	1247	459
1988	1125	521	105	1020	416
1989	1016	522	139	877	383
1990	862	587	98	764	489
1991	1452	972	263	1189	709
1992	1243	925	211	1032	714
1993	962	607	125	837	482
1994	1076	588	138	938	450
1995	736	484	91	645	393
1996	769	415	65	704	350
1997	840	437	87	753	350
1998	742	471	107	635	364
1999	804	436	102	702	334
2000	608	451	83	525	368
2001	763	865	140	623	725
2002	556	558	111	445	447
2003	502	463	96	406	367
2004	545	369	99	446	270
2005	474	277	90	384	187
2006	501	325	105	396	220
2007	505	282	77	428	205
2008	784	645	179	605	466
2009	988	883	312	676	571
2010	815	564	221	594	343
2011	462	480	116	346	364
Total	30787	18895	4290	26497	14605

Note: Cell entries are annual counts of articles retrieved for each Corpus. See text for details explaining the generation of each corpus. See footnote 27 for a description of the methods used to calculate article overlap.

Table 3: Proportion of Relevant Articles by Corpus

	Articles in both	Unique	Unique
Relevance	*SSW* and BBLMN	BBLMN	*SSW*
Relevant	0.44	0.42	0.37
Not Sure	0.00	0.00	0.00
Not Relevant	0.56	0.58	0.63

Note: Cell entries indicate the proportion of articles in each dataset (and their overlap) coded as providing information about how the US economy is doing. One thousand articles from each dataset were coded by three CrowdFlower workers located in the US. Each coder was assigned a weight based on her overall performance before computing the proportion of articles deemed relevant. If two out of three (weighted) coders concluded an article was relevant, the aggregate response is coded as "relevant".

Table 4: **Comparison of Inter-coder Reliability of Sentences Coded by Undergraduate Students and by Crowd Workers**

Coders	Variance	APIA	Sentences	Codings
Undergraduate Students (all)	0.30	0.82	1051	3254
Crowd workers (all)	0.57	0.74	1788	4227
Undergraduate Students (overlap)	0.28	0.84	420	1032
Crowd workers (overlap)	0.53	0.72	420	1226

Note: *Variance* is average variance of codings within sentences, *APIA* is the average pairwise inter-coder agreement, *sentences* indicates the total number of unique sentences coded, and *codings* indicates the number of relevant codings at the sentence level. Sentences coded as not relevant are excluded from the analysis. The first two rows correspond to *all* the sentences coded by undergraduate students and crowd workers in our dataset, respectively. The second two rows correspond to the subset of sentences coded by *both* students and crowd workers. This allows us to compare inter-coder reliability on the exact same dataset.

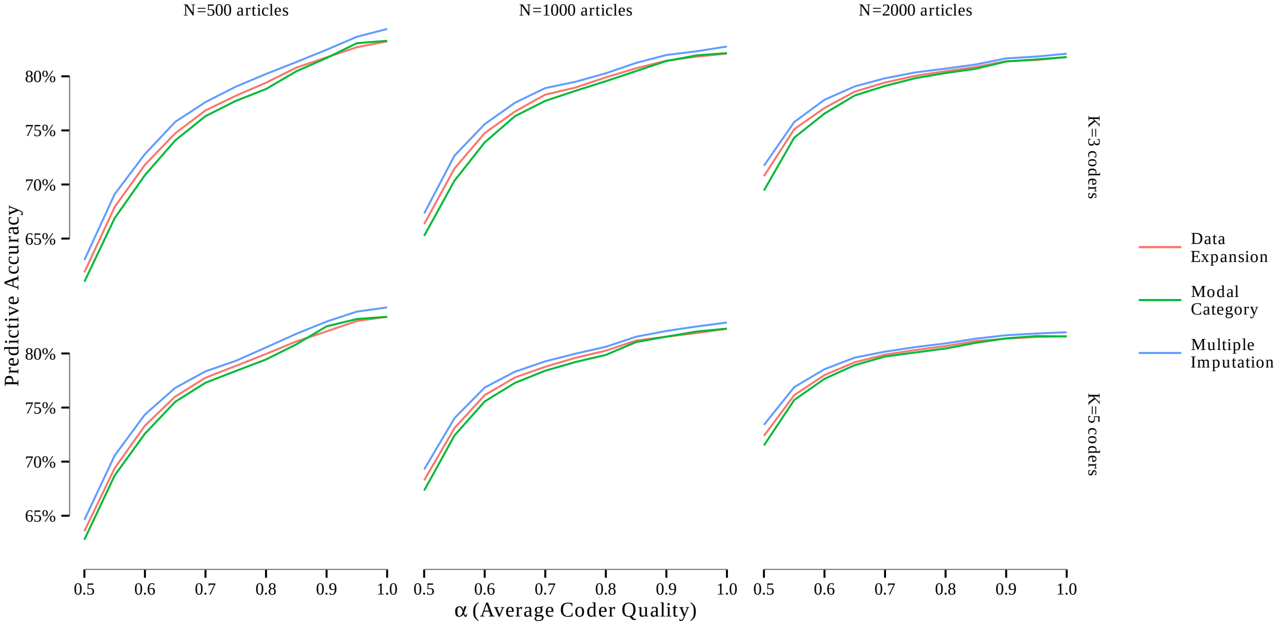
Table 5: **Top Predictive N-grams in Classifier**

Negative n-grams
declin, recess, cost, unemploy, their, slump, off, offic, fell, down, deficit, loss, drop, plung, washington, american, school, day, problem, hous, fear, presid, chief, case, anoth, system, adjust, much, peopl, friday, worst, lend, layoff, part, the most, limit, our, need, be, health

Positive n-grams
gain, strong, rise, growth, advanc, recoveri, year, januari, person, sale, earlier, rose, meet, incom, better, save, the fed, expect the, continu, into, improv, gold, manufactur, current, optim, also, activ, new york, increas, while, set, spend, three, york, market, good, two, survey, progress, payrol

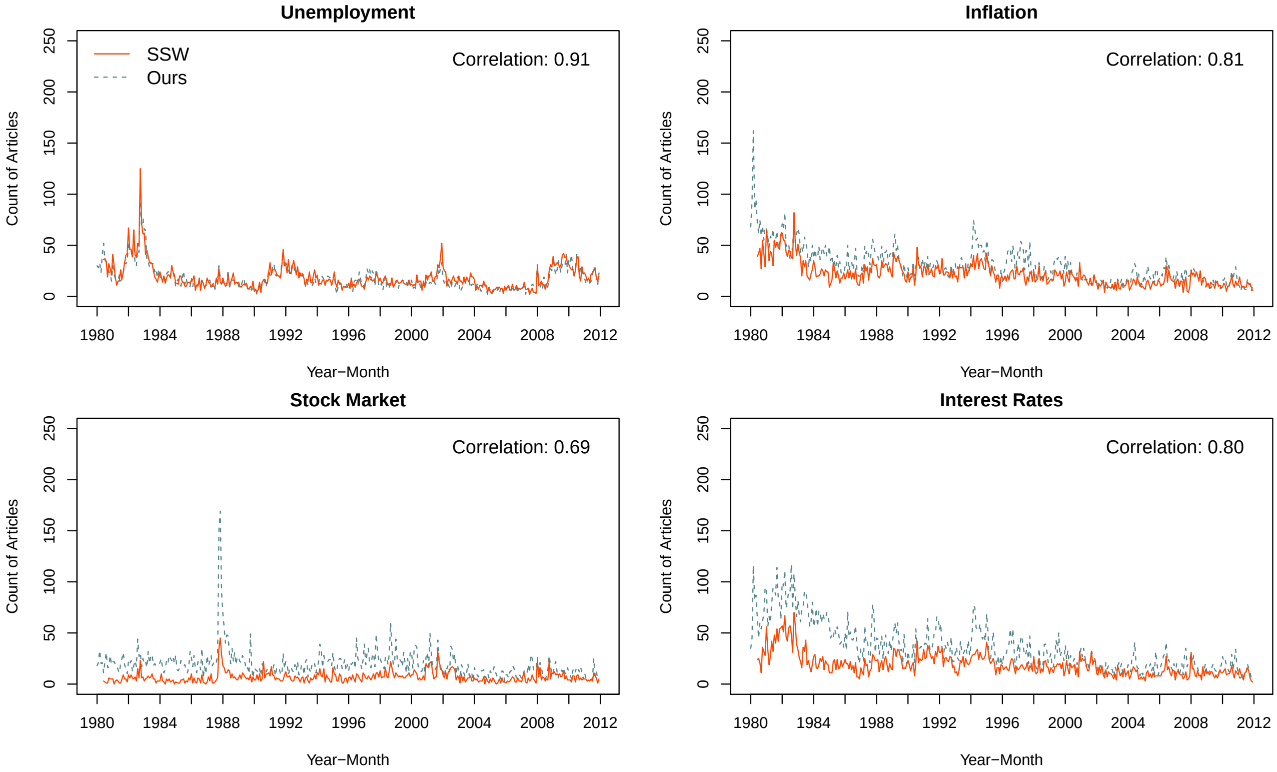
Note: These lists include the n-grams associated with the highest and lowest coefficients, as estimated by our machine learning classifier.

Figure 1: **Accuracy with Classifier Training Using Multiple Codings**



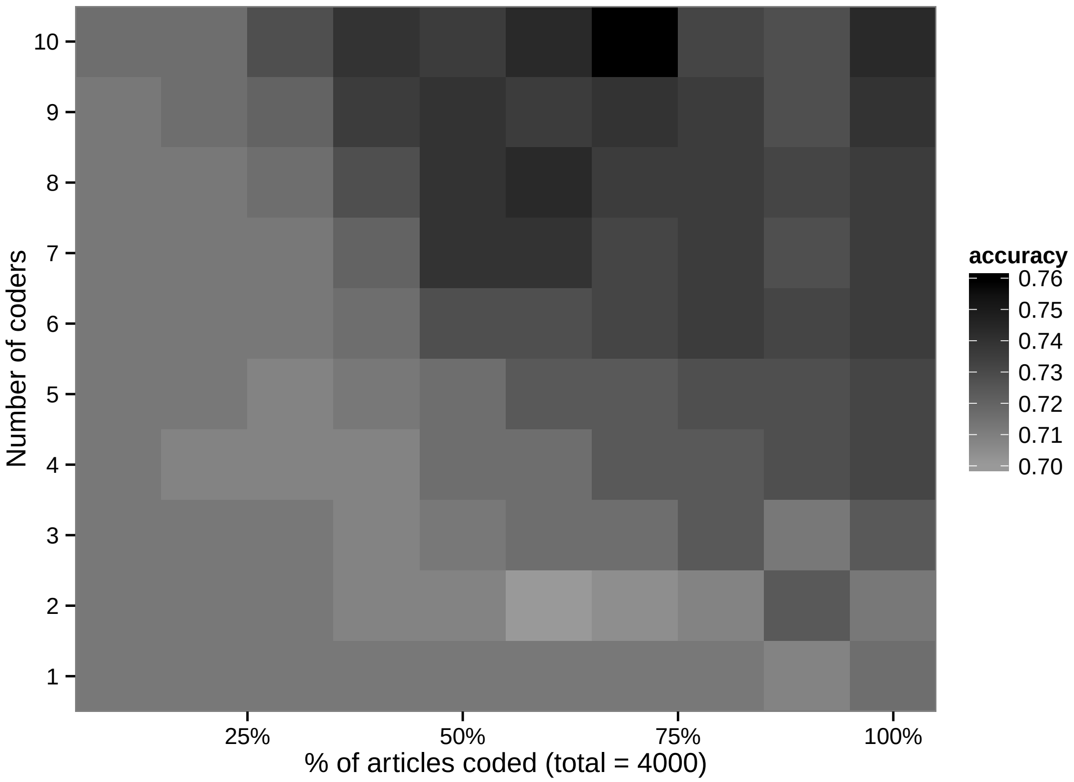
Note: Results based on a simulation in which we created a dependent variable y with a Bernoulli distribution about p, where $p = F(\mathbf{Xb} + \mathbf{e})$, $\mathbf{X}$ is a feature matrix with N rows and 100 columns, with $x_{ij} = \mathcal{N}(0,1)$, and $\mathbf{b}$ is a coefficient vector of [2, -2, 1, -1, 0, 0, 0 . . . , 0, 0, 0], of length 100, and $\mathbf{e}$ is random noise, with $\mathbf{e} = \mathcal{U}(-1,1)$. We create K coders of quality level α, where α is the percentage of objects that a coder of quality α codes correctly. Thus a coder of quality level 100 has 0 variance, they code every object correctly. We then simulated 100 draws of all possible cases with combinations of these parameters, and estimated a logit classifier on each draw *using each of our 3 methods for multiple codings*: use the modal coding, stack the coding, or use the multiple imputation method. Predictive accuracy reports the percentage of objects correctly classified as positive or negative given the true sentiment score, p.

Figure 2: **Comparison of Monthly Counts of Articles Containing Key Economic Terms in the *SSW* Corpus & in Our Corpus (*BBLMN*)**



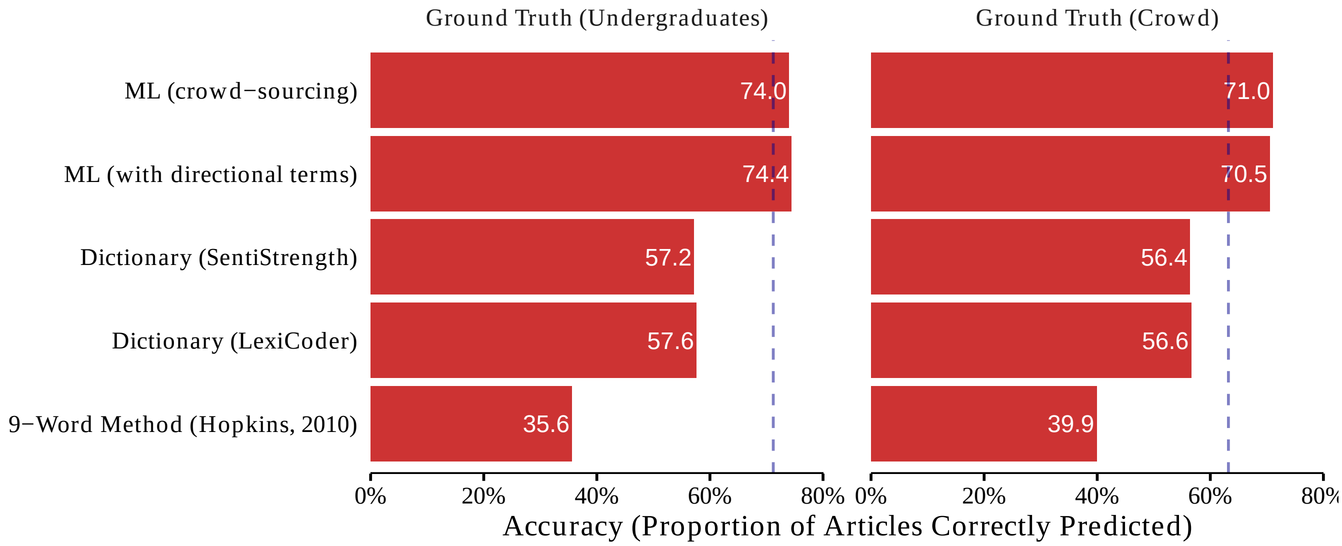
Note: Article counts in *SSW* are based on the dataset used in Soroka et al. (2015) using the search given in footnote 23 and post search filters described in section 2.1. Article counts in *BBLMN* are based on the search and post search filters described in section 2.1. Monthly counts report the total number of articles in the respective dataset that include the given n-gram in a given month.

Figure 3: **Heat Map of Accuracy by Number of Coders and Number of Articles**



Note: Shading of cells in the heat map indicate accuracy of a classifier trained with that specific number of coders and articles. Not relevant articles are excluded.

Figure 4: **Performance of Machine Learning and Dictionary Methods – Accuracy**



Note: Accuracy (percentage of articles correctly classified) is reported for the ground truth dataset coded by undergraduate students (left) and by 10 CrowdFlower coders (right). The dashed vertical lines indicate the baseline level of accuracy if the modal category is always predicted.