An Improved Method of Automated Nonparametric Content Analysis for Social Science*

Connor T. Jerzak† Gary King‡ Anton Strezhnev§

February 7, 2019

Abstract

Computer scientists and statisticians often try to classify individual textual documents into chosen categories. In contrast, social scientists more commonly focus on populations and thus estimate the proportion of documents falling in each category. The two existing types of techniques for estimating these category proportions are parametric "classify and count" methods and "direct" nonparametric estimation of category proportions without an individual classification step. Unfortunately, classify and count methods can sometimes be highly model dependent or generate more bias in the proportions even as the percent correctly classified increases. Direct estimation avoids these problems, but can suffer when the meaning and usage of language is too similar across categories or too different between training and test sets. We develop an improved direct estimation approach without these issues by introducing continuously valued text features optimized for this problem, along with a form of matching adapted from the causal inference literature. We evaluate our approach in analyses of a diverse collection of 73 data sets, showing that it substantially improves performance compared to existing approaches. As a companion to this paper, we offer easy-to-use software that implements all ideas discussed herein.

*Our thanks to Neal Beck, Aykut Firat, and Ying Lu for data and helpful comments.

†PhD Candidate and Carl J. Friedrich Fellow, 1737 Cambridge Street, Harvard University, Cambridge MA 02138, ConnorJerzak.com, cjerzak@g.harvard.edu.

‡Albert J. Weatherhead III University Professor, Institute for Quantitative Social Science, 1737 Cambridge Street, Harvard University, Cambridge MA 02138; GaryKing.org, King@Harvard.edu, (617) 500-7570.

§PhD Candidate, 1737 Cambridge Street, Harvard University, Cambridge MA 02138, antonstrezhnev.com, astrezhnev@fas.harvard.edu.

1 Introduction

One of the defining characteristics of social science is a focus on population-level generalizations. We discover an interesting puzzle about one election, but try to develop theories that apply to many more. We are interested in the politics of one country, but attempt to understand it as an example of how all countries (or all democracies, or all developing countries, etc.) operate. We survey 1,500 Americans about their political attitudes, but we seek to understand how all Americans, or all people, form attitudes. Social scientists usually leave to journalists whether any one congressional speech supports a particular policy and focus instead on the percent of all speeches that support the policy.

Different emphases across disciplinary areas generate divergent methodological approaches. For example, computer scientists and statisticians often focus on the classification of an individual object (web page, war, person, document, country, social media post, etc.) into a set of mutually exclusive and exhaustive categories. Social scientists use classification methods too, but their interest is more often on aggregate generalizations about populations of objects, such as the percent in each category, rather than any one individual classification, a task that is sometimes called "quantification".[1]

Applying a simple "Classify and Count" approach yields accurate category percentages under a perfect classifier. Perfect classifiers are unrealistic in real applications (Hand, 2006), but they are unnecessary for aggregate accuracy if individual-level errors cancel. Moreover, choosing a classifier by maximizing the percent correctly classified can sometimes drastically increase the bias of aggregate quantities. For example, the decision rule "war never occurs" accurately classifies country-year dyads into war/no war categories with over 99% accuracy, but is obviously misleading for political science research.

Similarly, the proportion of email you receive that lands in your spam folder is a biased estimate of the percent of spam you receive overall because spam filters are tuned to

[1] Estimating category percentages, as opposed to individual classifications, is also of interest in epidemiology, where it is called "prevalence estimation." Interest in the technical area is also growing in computer science, machine learning, computational linguistics, and data mining, where it is variously called "quantification," "class prior estimation," "counting," "class probability re-estimation," and "learning of class balance." See Buck and Gart (1966), Levy and Kass (1970), Forman (2007), Milli et al. (2013), Esuli and Sebastiani (2015), Kar et al. (2016), and Tasche (2016) and the unification in Firat (2016).

the fact that people are more annoyed when they miss an important email than when some spam appears in their inbox. This is easy to fix by tuning your spam filter to avoid the bias, or correcting after the fact, but in most applications we do not know the classifier's bias. To be more specific, a method that classifies 60% of documents correctly into one of 8 categories might be judged successful and useful for classification. For example, if Google or Bing were to provide relevant results in 60% of searches (which is about the average empirically), we might be quite satisfied since the low cost of misclassification is to merely choose another search term and try again. However, because the individual category percentages can then be off by as much as 40 percentage points, the same classifier may be less useful for social science.

The tasks of estimating category percentages (quantification) or classifying individual documents (classification) both begin by analyzing a small subset of documents with (usually hand-coded) category labels. Classification methods normally require these labeled and unlabeled document sets to be drawn from the same population, so the class probabilities can be calibrated. Commonly, however, the labeled set is created in one time period and a sequence of unlabeled sets are collected during subsequent time periods, each with potentially different distributions. For example, scholars may hand-label a set of social media posts about a presidential candidate into the 10 reasons people like or do not like this person. Then, for each day after the hand coding, a researcher may try to estimate the percent of posts in each of these categories using the initial hand-labeled set, with no new coding of documents. The methods of quantification we discuss here are designed to accommodate these situations even though these are the circumstances where the assumptions behind classification methods are violated.

We build on the only nonparametric quantification method developed for estimating multi-category proportions that does not resort to individual classification as a first step. This methodology was developed in King and Lu (2008), with survey research applications in public health, and in Hopkins and King (2010), with applications to text analysis in political science; and it was extended in King, Lu, and Shibuya (2010) and King, Pan, and Roberts (2013, Appendix B). A U.S. Patent has been issued for the technology (King,

Hopkins, and Lu, 2012) and licensed by a university to a firm originally formed to implement an industrial strength version (Crimson Hexagon). Over 2,000 scholarly articles in several scholarly fields have cited these works (according to Google scholar). The method has come to be known by the name "readme," which is the widely-used free open source software that implements it (Hopkins, King, Knowles, and Melendez, 2013).

We begin by developing the intuition behind readme's nonparametric methodology, and highlight situations where it can perform poorly. We then outline an approach for improving performance via two techniques, both of which involve better representing the meaning of the text. First, our technique allows for changes in the meaning and use of language over time by adapting matching techniques developed from the causal inference literature. (We also show in Appendix C how the methods we developed may even contribute something back to the causal inference literature.) Second, we develop an algorithm that chooses a feature space to discriminate between categories with as many non-redundant or independent features as possible. Unlike principal components analysis, independent component analysis, random projections, t-distributed stochastic neighborhood embeddings, or others designed for exploration, visualization, or classification, our approach is the first to generate a feature space optimized for quantification.

We summarize the readme estimator and its key assumptions in Section 2. In Section 3, we use analytical and simulation analyses to specify factors that affect the bias and variance of readme. Section 4 then introduces our new methodology. In Section 5, we compare our approach to readme in out-of-sample empirical evaluations in 19,710 data sets, derived from subsets of 73 corpora (and repeated with 18 different evaluation protocols). We discuss what can go wrong and how to avoid it in Section 6. We do not claim that our approach will perform better than readme or other methods in all data sets; the well-known "ping pong theorem" shows this is impossible (i.e., every method includes enough tweakable options that any contest among these methods is usually won by the researcher who goes last; Hoadley 2001). Our more specific claim is that our approach will normally outperform other approaches in real data, under the real-world conditions we describe below. The results are encouraging. Section 7 concludes; mathematical proofs

and evaluation protocols appear in the appendix.

2 Readme: Estimation without Classification

We now describe readme in a manner that conveys its logic, while also laying the groundwork for our subsequent improvements. The technique begins with a *text-to-numbers* step, that maps the entire set of textual documents into a numerical feature space. In the second *estimation* step, we apply a statistical method to summaries of the numerical feature space for the purpose of estimating the category proportions of interest. Our running example is textual documents, where humans hand code labels for documents, but the methodology also applies to any other set of objects (such as people, deaths, attitudes, buildings, books, etc.) for which the goal is estimating category proportions.

Notation Consider two sets of textual documents — L, which includes N^{L} documents *labeled* with a category number, and U, which includes N^{U} *unlabeled* documents — where $N = N^{\mathrm{L}} + N^{\mathrm{U}}$. When there is no ambiguity, we use i as a generic index for a document in either set and N as a generic description of either set size. Each document falls into category c in a set of mutually exclusive and exhaustive categories ($c \in \{1, \ldots, C\}$), but the category label is only observed in the labeled set. We write $D_i = c$ to denote that document i falls into category c. Denote $N_c^{\mathrm{L}} = \sum_{i=1}^{N^{\mathrm{L}}} \mathbf{1}(D_i = c)$ as the number of documents in category c in the labeled set, N_c^{U} as the (unobserved) number in c in the unlabeled set, and N_c generically for either set in category c.

The proportion of unlabeled documents in category c is $\pi_c^{\mathrm{U}} = \mathrm{mean}_{i \in U}[\mathbf{1}(D_i = c)]$ (where for set A with cardinality $\#A$, the mean over i of function $g(i)$ is $\mathrm{mean}_{i \in A}[g(i)] = \frac{1}{\#A} \sum_{i=1}^{\#A} g(i)$). The vector of proportions $\pi^{\mathrm{U}} \equiv \{\pi_1^{\mathrm{U}}, \ldots, \pi_C^{\mathrm{U}}\}$, which represents our quantity of interest, forms a simplex, i.e. $\pi_c^{\mathrm{U}} \in [0, 1]$ for all c and $\sum_{c=1}^{C} \pi_c^{\mathrm{U}} = 1$. We also define the analogous (but observed) category proportions for the labeled set π^{L}.

Text to Numbers In this first step, we map the entire labeled and unlabeled corpora, with the document as the unit of analysis, into a constructed space of textual features. Many ways of performing this mapping can be created, and we propose a new one below

optimized for quantification. For readme, Hopkins and King (2010) began with a set of k unigrams, each a binary indicator for the presence (coded 1) or absence (0) of a chosen word or word stem in a document. The number of possible strings of these zeros and ones, called a *word stem profile*, is $W = 2^k$.

The readme approach then computes a W-length *feature vector* S^{L} by sorting the labeled documents into the W mutually exclusive and exhaustive word stem profiles, and computing the proportion of documents that fall in each. To make the definition of $S^{\mathrm{L}} = \{S^{\mathrm{L}}_w\}$ more precise and easier to generalize later, begin with the $N^{\mathrm{L}} \times W$ *document-feature matrix* $F = \{F_{iw}\}$ with rows for documents, and columns for features which in this case are unique word stem profiles. Each element of this matrix, F_{iw}, is a binary indicator for whether document i is characterized by word stem profile w. Then elements of S^{L} are column means of F: $S^{\mathrm{L}}_w = \mathrm{mean}_{i \in L}(F_{iw})$. Then the same procedure is applied, with the same word stem profiles, to the unlabeled set, which we denote S^{U}.

We also define a W-length *conditional feature vector* as $X^{\mathrm{L}}_c = \{X^{\mathrm{L}}_{wc}\}$, which results from the application of the same procedure within category c in the labeled set, and $X^{\mathrm{U}}_c = \{X^{\mathrm{U}}_{wc}\}$ within category c in the unlabeled set (X^{U}_c is unobserved because c is unknown in the unlabeled set). These can be computed from F^c, a document-feature matrix representing only documents in category c. We then collect these vectors for all categories into two $W \times C$ matricies, $X^{\mathrm{L}} = \{X^{\mathrm{L}}_1, \ldots, X^{\mathrm{L}}_C\}$ and $X^{\mathrm{U}} = \{X^{\mathrm{U}}_1, \ldots, X^{\mathrm{U}}_C\}$, respectively.

Estimation Our goal is to estimate the vector of unlabeled set category proportions $\pi^{\mathrm{U}} = \{\pi^{\mathrm{U}}_1, \ldots, \pi^{\mathrm{U}}_C\}$ given S^{L}, S^{U}, and X^{L}. Begin with an accounting identity (i.e., true by definition), $S^{\mathrm{U}}_w = \sum_{c=1}^{C} X^{\mathrm{U}}_{wc} \pi^{\mathrm{U}}_c, \ \forall \ w$, or equivalently in matrix form:

$$S^{\mathrm{U}} = X^{\mathrm{U}} \pi^{\mathrm{U}}. \tag{1}$$

This is an application of the Law of Total Probability, which is a fundamental rule relating marginal and conditional probabilities. The key for quantification purposes is that we can solve this expression for the quantity of interest as in linear regression, $\pi^{\mathrm{U}} = (X^{\mathrm{U}\prime} X^{\mathrm{U}})^{-1} X^{\mathrm{U}\prime} S^{\mathrm{U}}$. However, π^{U} cannot be directly computed this way since we do not observe the "regressor" X^{U}. So instead readme estimates π^{U} by using X^{L}, which

is observed in the labeled set and yields $\widehat{\pi^{\mathrm{U}}} = (X^{\mathrm{L}\prime}X^{\mathrm{L}})^{-1}X^{\mathrm{L}\prime}S^{\mathrm{U}}$ (or a modified version of this expression that explicitly preserves the simplex constraint).

Readme works with the above estimator with any choice of k word stems. Hopkins and King (2010) then randomly select many subsets of $k \approx 16$ word stems, run the algorithm for each, and average the results. This step is one way to reduce the dimensionality of the text. By averaging across word stem profiles, the variance of the final estimator is also reduced. Alternatively, we can use this estimator with all features simultaneously with a different computational algorithm (Ceron, Curini, and Iacus, 2016). We return to this step and improve it in Section 4.

Assumptions First, since the unlabeled conditional feature matrix X^{U} is unobserved, Hopkins and King (2010) assume $X^{\mathrm{U}} = X^{\mathrm{L}}$. However, it turns out that this assumption, as stated, is unnecessarily restrictive. In fact, we can get the same statistical result by expressing the labeled conditional feature matrix as an unbiased and consistent estimator of the unlabeled conditional feature matrix:

$$E(X^{\mathrm{L}}) = X^{\mathrm{U}}, \qquad \lim_{N^{\mathrm{L}}\to\infty} X^{\mathrm{L}} = X^{\mathrm{U}}. \tag{2}$$

Assumption 2 about the *conditional* distribution of features and categories is considerably weaker than that made by classifiers, which is that (a) the *joint* distribution of features and categories is the same in the labeled and unlabeled sets, (b) the measured features span the space of all predictors of D, and (c) the estimated model nests the true model as a special case (Hand, 2006). Because the correct model linking features to categories is unknown ex ante, this assumption is difficult to satisfy. On the contrary, readme does not need to assume a model for S since the relationship between the unconditional and conditional feature vectors follows directly from the laws of probability applied in Equation 1. (Assumption 2 can be violated due to semantic change, which can happen if the labeled set is hand coded at one time and the unlabeled set is collected at another time or in another place for which the meaning of language differs. We weaken this assumption further, allowing semantic change, below.)

Second, to ensure $\widehat{\pi^{\mathrm{U}}}$ is uniquely defined, we must assume the matrix X^{L} is of full

rank, which translates into (a) feature choices that lead to $W > C$ and (b) the lack of perfect collinearity among the columns of X^{L}. Assumption (a) is easy to control by generating a sufficient number of features from the text. Assumption (b) is only violated if the feature distributions in documents across different categories are identical, which is unlikely with a sufficient number of coded documents. (We prove in Section 3 that high collinearity, which can result if categories are weakly connected to the features or documents are labeled with error, can exacerbate the bias of the readme estimator, which provides an important clue to generate improvements.)

3 Statistical Properties

Our goal is to understand the situations where readme performs poorly so we can design improvements (in Section 4). We show here, through analytical calculations (Section 3.1) and simulations (Section 3.2), that three situations can degrade readme performance.

First is *semantic change*, which is the difference in the meaning of language between the labeled and unlabeled sets. Authors and speakers frequently morph the semantic content of their prose to be clever, get attention, be expressive, curry political favor, evade detection, persuade, or rally the masses. For these or other purposes, the content, form, style, and meaning of every symbol, object, or action in human language can always be contested.

We address two types of semantic change that impact readme: *emergent discourse*, where new words and phrases, or the meanings of existing words and phrases, appear in the unlabeled set but not the labeled set, and *vanishing discourse*, where the words, phrases, and their meanings exist in the labeled set but not the unlabeled set. "Russian election hacking" following the 2016 US presidential election is an example of emergent discourse, language which did not exist a few years before, whereas "Russian Communism" is an example of vanishing discourse, a usage that has been disappearing in recent decades. However, emergent and vanishing discourse can reverse their meanings if the researcher swaps which set is labeled. For example, in analyzing a large historical data set, a researcher may find it more convenient to read and label a contemporary data set

and infer to the historical data sets (e.g., as they are recovered from an archive); to label documents at the start of the period and infer to subsequent periods; or to code a sample spread throughout the period and to infer to the full data set. Either vanishing or emergent discourse can bias readme, but only if such discourse is present in a specific way (we describe below) across the categories. We show how to reduce bias due to vanishing discourse in Section 4.3.

Second is the *lack of textual discrimination*, where the language used in documents falling in different categories or across features is not clearly distinguishable. For clarity, we divide textual discrimination into two separate concepts that, in the readme regression, are related to the minimal requirement in least squares that, to reduce variance and model dependence, X must be of full rank and, ideally, as far as possible from degeneracy. Since X represents categories as variables and features of the text as rows, we refer to these two variables as *category distinctiveness* and *feature distinctiveness*, respectively.

The lack of textual discrimination may arise because the conceptual ideas underlying the chosen categories or features are not distinct. Hand coding errors can also lead to this problem, which is commonly revealed by low levels of intercoder reliability. The problem can also occur because of heterogeneity in how authors express information or a divergence between how authors of the documents express this information and how the analyst conceptualizes the categories or codes the features. We have also seen many data sets where the analyst begins with distinct and well-defined conceptual definitions for the set of C categories, with examples of documents that fall unambiguously into each one, but where it turns out upon large-scale coding that large numbers of documents can only be described as falling into multiple categories. Adding categories to represent these more complicated expressions (so that the resulting set is still mutually exclusive and exhaustive) is a logical solution, but this step often leads to a more cognitively demanding hand coding problem that reduces inter-coder reliability.

A final problematic situation for readme occurs due to interactions with the other two problems. This issue is *proportion divergence*, when the category proportions in the labeled set (π^{L}) diverge from those in the unlabeled set (π^{U}). To understand this issue,

consider a data set with massive semantic change and no textual discrimination — so the document texts are largely uninformative — but where $E(\pi^{\mathrm{L}}) = \pi^{\mathrm{U}}$, such as occurs when the labeled set is a random sample from the test set. In this situation, readme will return the observed proportion vector in the labeled set, π^{L}, which is an unbiased estimate of π^{U}. This means that we can sometimes protect ourselves from semantic change and the lack of textual discrimination by selecting a labeled set with a similar set of category proportions as the unlabeled set. This protective measure is impossible to put into practice in general, as it requires *a priori* knowledge of category membership, but it can be useful in some cases when designing training sets and we show below that it can be corrected for.

The rest of this section provides the analytical and simulation evidence to support, clarify, and further articulate these three situations where readme can be improved.

3.1 Analytical Results

In the classic errors-in-variables linear regression model, with random measurement error only in the explanatory variables, least squares is biased and inconsistent. Under Assumption 2, readme is also a linear regression with random measurement error in the explanatory variables. However, the readme regression is computed in the space of features, the size of which remains constant as the number of observations grows. As such, readme *is* statistically consistent: as we gather and code more documents for the labeled set (and keep W fixed, or at least growing slower than n), its estimator converges to the truth: $\lim_{N^{\mathrm{L}}\to\infty} \widehat{\pi^{\mathrm{U}}} = \lim_{N^{\mathrm{L}}\to\infty} (X^{\mathrm{L}\prime}X^{\mathrm{L}})^{-1}X^{\mathrm{L}\prime}S^{\mathrm{U}} = (X^{\mathrm{U}\prime}X^{\mathrm{U}})^{-1}X^{\mathrm{U}\prime}S^{\mathrm{U}} = \pi^{\mathrm{U}}$. This is a useful result suggesting that, unlike classic errors-in-variables, labeling more observations can reduce bias and variance. It also suggests that, to improve readme, we should focus on finite sample bias rather than consistency, which is already guaranteed.

To analyze readme's bias in a way that will provide useful intuition, consider a simplified case with only two categories. Because of the simplex constraint, the unlabeled set category proportions can be characterized by a single parameter, π_1^{U}, and the accounting identity for each feature mean w, S_w^{U}, can be written simply as:

$$S_w^{\mathrm{U}} = X_{w2}^{\mathrm{U}} + B_w^{\mathrm{U}}\pi_1^{\mathrm{U}} \tag{3}$$

where $B_w^{\mathrm{U}} = X_{w1}^{\mathrm{U}} - X_{w2}^{\mathrm{U}}$. The readme estimator is then the least-squares estimator of π_1^{U}, which we write as follows. (Proofs of all propositions appear in Appendix A.)

Proposition 1. *Two-category readme estimator is*

$$\widehat{\pi_1^U} = \frac{\sum_{w=1}^{W} B_w^L \left(S_w^U - X_{w2}^L\right)}{\sum_{w=1}^{W} (B_w^L)^2}, \tag{4}$$

where $B_w^L = X_{w1}^L - X_{w2}^L$.

If $X^{\mathrm{L}} = X^{\mathrm{U}}$, the above expression equals π_1^{U} and readme is unbiased. However, due to sampling error, the realized sample value of X^{L} may differ from the unobserved true value X^{U}. By Assumption 2, $X_{wc}^{\mathrm{L}} = X_{wc}^{\mathrm{U}} + \epsilon_{wc}$ where ϵ_{wc} is a random variable with mean zero and variance inversely proportional to N_c. This enables us to write the readme estimator in terms of X^{U}, the true unlabeled set category proportion π_1^{U}, and the sample category size N_c^{L}. The next proposition gives the expectation of this quantity.

Proposition 2. *The expected value of the two-category readme estimator is*

$$\mathbb{E}\left[\widehat{\pi_1^U}\right] = E\left[\frac{\sum_{w=1}^{W} \left(B_w^U + \nu_w\right) B_w^U}{\sum_{w=1}^{W} \left(B_w^U + \nu_w\right)^2}\right] \pi_1^U - E\left[\frac{\sum_{w=1}^{W} \left(B_w^U + \nu_w\right) \epsilon_{w2}}{\sum_{w=1}^{W} \left(B_w^U + \nu_w\right)^2}\right]. \tag{5}$$

where $B_w^U = X_{w1}^U - X_{w2}^U$ *and* $\nu_w = \epsilon_{w1} - \epsilon_{w2}$.

The consistency property of readme can be seen here: As the error in measuring X^{U} with X^{L} goes to 0 or N^{L} goes to infinity, the second term in the expectation is 0 (because $\epsilon_{w2} \propto 1/N_C^{\mathrm{L}} \to 0$), while the first converges to π_1^{U}. In the presence of measurement error, the bias is a function of two components of the lack of textual discrimination — (a) the difference in the true category proportions, $B_w^{\mathrm{U}} = X_{w1}^{\mathrm{U}} - X_{w2}^{\mathrm{U}}$ and (b) the combined error variance $\nu_w = \epsilon_{w1} - \epsilon_{w2}$.

We obtain further intuition via an approximation using a first-order Taylor polynomial:

Proposition 3. *The approximate bias of the readme estimator is*

$$\textit{Bias}\left(\widehat{\pi_1^U}\right) \approx \frac{\sum_{w=1}^{W} \left[\textit{Var}(\epsilon_{w2}) - \textit{Cov}(\epsilon_{w1}, \epsilon_{w2})\right] \left(1 - \pi_1^U\right) - \left[\textit{Var}(\epsilon_{w1}) - \textit{Cov}(\epsilon_{w1}, \epsilon_{w2})\right] \pi_1^U}{\sum_{w=1}^{W} \left[(B_w^U)^2 + \textit{Var}(\nu_w)\right]}. \tag{6}$$

This expression suggests four insights. First, as the systematic component of textual discrimination, B_w^{U}, increases relative to the variance of the error terms, ϵ_{w1} and ϵ_{w2}, the bias

approaches 0. In other words, readme works better when the language of the documents across categories is distinct.

Second, adding more informative numerical representations of the text, so that W increases (but with a fixed n), has an indeterminate impact on the bias. While more informative numerical summaries of the text can increase the sum in the denominator, they may increase the overall bias if the result is an error variance that is high relative to the discriminatory power.

Third, we confirm the intuition that larger labeled sets within each category are better: Since the elements of X^{L} are simple means across documents assumed to be independent, the variance of the measurement error terms is simply $\mathrm{V}(\epsilon_{wc}) = \sigma^2_{wc}/N^{\mathrm{L}}_c$, which decline as the labeled set category sizes increase.

Finally, we simplify by studying the special case of independence of measurement errors across categories (i.e. $\mathrm{Cov}(\epsilon_{w1}, \epsilon_{w2}) = 0$). In this situation, readme bias is minimized when the following relationship holds between the labeled and unlabeled set category proportions:

Proposition 4. *When measurement errors are independent across categories, the bias of readme is minimized at*

$$\pi^L_1 = \frac{\pi^U_1 \sum_{w=1}^{W} \sigma^2_{w1}}{\pi^U_1 \sum_{w=1}^{W} \sigma^2_{w1} + (1 - \pi^U_1) \sum_{w=1}^{W} \sigma^2_{w2}} \tag{7}$$

What this means is that when measurement error variances are roughly equivalent across categories, the bias of readme is smallest when category proportion divergence is smallest.

We will use the intuition from each of these results in Section 4.

3.2 Simulation Results

We can illustrate these analytical results using simulation to further build intuition for Section 4. We have found that simulations with large values of C and W generate more complexity without much additional insight, and so for expository purposes in this section we set $C = 2$ and $W = 2$. We then evaluate readme performance as we vary the degree of semantic change, textual discrimination, and proportion divergence. Our improvements to readme exploit the relationship between these three quantities.

For the purpose of our simulations, define category distinctiveness, or columns of X, as $(b_1 + b_2)/2$, where the absolute differences between categories is $b_w = |X_{wc} - X_{wc'}|$ for row $w = 1, 2$. Then, we define feature distinctiveness as $|b_1 - b_2|/2$, which is is the distinctiveness between rows of X. (We show how each definition generalizes with $C > 2$ and $W > 2$ in Section 4.2.) For our error metric, we use the sum of the absolute errors over categories (SAE), averaged over simulations.[2]

Figure 1 illustrates how the SAE behaves as a function of category distinctiveness (vertically) and proportion divergence (horizontally). SAE is coded in colors from low (blue) to high (yellow). The bottom right of the figure is where readme performance is best: where proportion divergence is low and category distinctiveness is high. When the language is clearly distinguishable among categories, readme can overcome even large divergences between the labeled and unlabeled sets. Without high levels of textual discrimination, readme then becomes vulnerable to high levels of proportion divergence. Category distinctiveness and proportion divergence appear to have roughly the same relative importance, as the contour lines in Figure 1 are not far from 45° angles.

Figure 2 studies textual discrimination further by illustrating how category distinctiveness (horizontally) and feature distinctiveness (vertically) jointly impact SAE. If we hold feature distinctiveness fixed, increased category distinctiveness improves performance; if we hold category distinctiveness fixed, greater feature distinctiveness similarly leads to better performance over most of the range. Of the two, feature distinctiveness is somewhat more predictive of performance, especially for low levels of category distinctiveness.

Finally, Figure 3 illustrates how the relationship between feature distinctiveness (three

[2]We use the SAE to make our analysis consistent across data sets with different numbers of categories. We have found that simple attempts to normalize, such as dividing by the number of categories, tends to weaken this comparability, especially because in all cases the target quantity is located on the simplex. We draw our simulations as follows. First, we control proportion divergence by drawing π^{L} and π^{U} from i.i.d. Dirichlet distributions with concentration parameters set to 1. (In our figures, we measure average proportion divergence as $(|\pi_1^{\mathrm{L}} - \pi_1^{\mathrm{U}}| + |\pi_2^{\mathrm{L}} - \pi_2^{\mathrm{U}}|)/2$.) We sample X_{wc}^{U} from an i.i.d. Normal with mean 0 and variance $1/9$. Then, we generate $X_{wc}^{\mathrm{L}} = X_{wc}^{\mathrm{U}} + \epsilon$, where $\epsilon = 0$ or, to simulate semantic change, from a Normal with a mean at 0 and standard deviation proportional to $|X_{wc}^{\mathrm{U}}|$. We then treat these parameters as fixed and, to simulate measurement error in the calculation of X^L, generate 5,000 repeated sample data sets from each set of parameters, apply the readme estimator, and estimate the mean over simulations of SAE. To generate each of the 5,000 sampled data sets, we randomly generate document-level features from Normal densities by adding a draw from a standard Normal to each cell value of X_{wc}^{L} and X_{wc}^{U}.

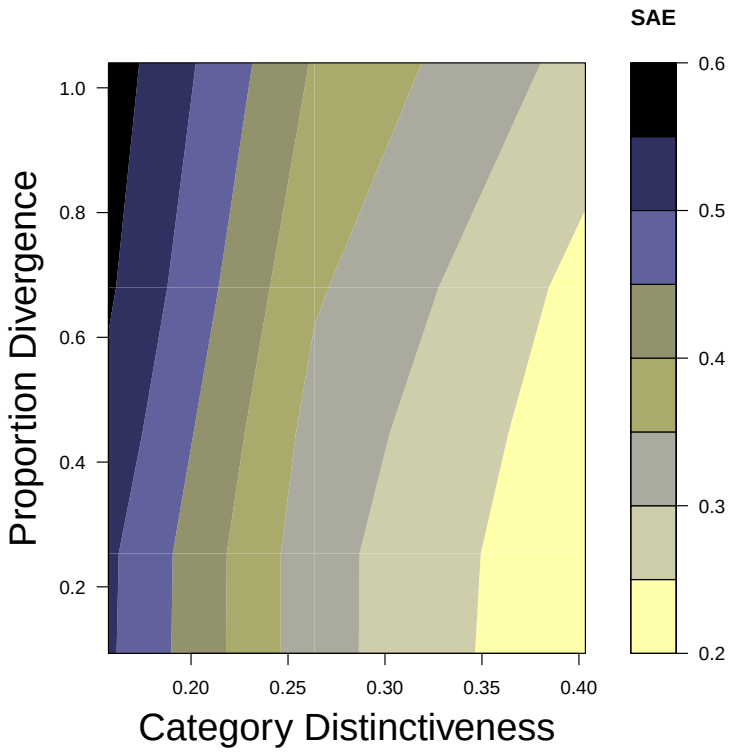

Figure 1: Category distinctiveness is plotted (horizontally) by proportion divergence between the labeled and unlabeled sets is plotted (vertically), with mean absolute sum of errors color coded (with yellow in the lower right corner best).

separate lines in each panel) and proportion divergence (horizontal axis) is mediated by the presence of semantic change (difference between the panels). Without semantic change (left panel), highly distinctive features greatly reduce SAE (which can be seen by the wide separation among the lines on the graph). In contrast, in the presence of semantic change (in this case we moved the mean of $E(X^{\mathrm{L}})$ by a quarter of a standard deviation from X^{U}), more distinctive features still tend to outperform less distinctive features, but the difference is less pronounced. With semantic change, distinctive features in the labeled set may no longer be distinctive in the unlabeled set or may in fact be misleading about documents' category membership.

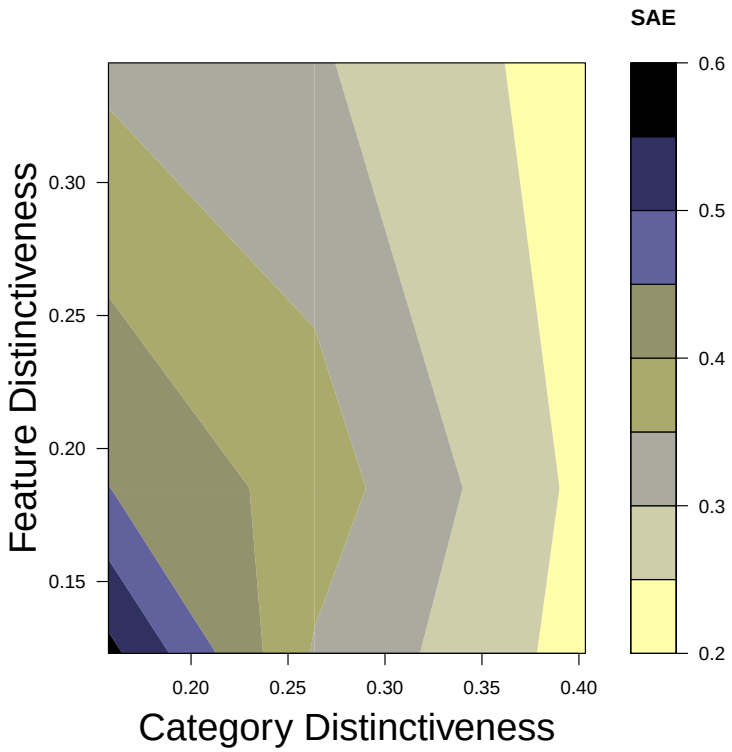

Figure 2: Category distinctiveness is plotted (horizontally) and feature distinctiveness (vertically), with mean absolute sum of errors color coded (with yellow indicating best performance).

4 Improvements

Section 3 offers analytical and simulation results to show how proportion divergence, textual discrimination (including category and feature distinctiveness), and semantic change impact the performance of readme. We now use these insights to develop a better method, which we call readme2, by optimizing the space of input features (Section 4.1), improving feature discrimination (Section 4.2), and correcting for semantic change and proportion divergence (Section 4.3).

4.1 Choosing a space of input features

Since numerical representation of text documents can have an outsized impact on the results of a particular method of estimation (see Denny and Spirling, 2016; Levy, Goldberg, and Dagan, 2015), we design a numerical representation optimized for the quantification

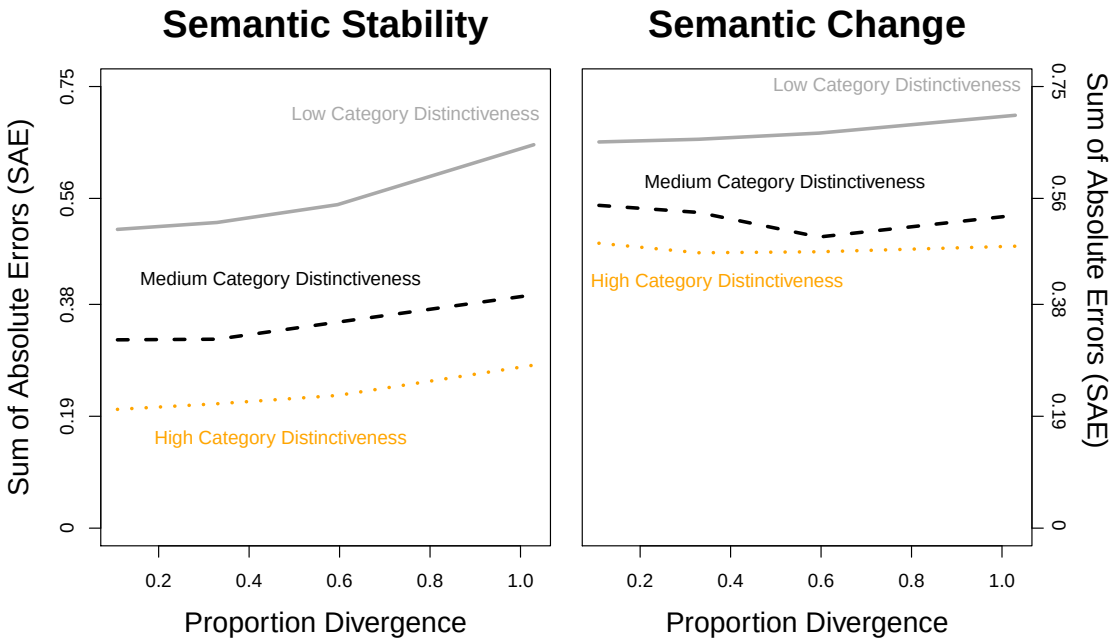

Figure 3: Proportion divergence is plotted (horizontally) by the mean of the sum of the absolute error (vertically) for different levels of category distinctiveness (separate lines) and for the absence (left panel) and presence (right panel) of semantic change.

problem. Hopkins and King (2010) treated documents as having a *single* "feature" – each document could belong to a single, mutually exclusive "word stem profile" based on the unique vector of word stems that appear in the document. While this approach could yield feature vectors with high degrees of discrimination by picking up on particular interactions between terms that happened to separate categories well, it has two undesirable properties — *inefficiency* and *sparsity*.

Readme's numerical representation of text is inefficient, in that documents that are identical except for a single word stem are treated as having zero features in common. And the readme representation can be sparse in that text corpora often have thousands of unique word stems, even after pre-processing, and so the chances of having more than a single document per word stem profile are minimal, resulting in little feature overlap between labeled and unlabeled sets. The sparsity of the feature space was solved in readme via random sub-sampling small numbers of individual word stems (in separate batches) to create word stem profiles, meaning that only a small fraction of the terms in a given

corpus would be used to construct the feature matrices used in estimation. Even with subsampling, some word stem profiles appearing in the unlabeled set may not appear at all in the labeled set, which requires that they be dropped from the analysis, which is either inefficient or changes the estimand.

An alternative approach to sparsity is to represent each term with one feature, such as "term frequency," perhaps normalized. This has the advantage over word stem profiles of guaranteeing some overlap in the distributions of features between labeled and unlabeled sets. However, for most applications, the vast majority of terms in a corpus have limited discriminatory power across categories. Since readme relies heavily on features whose distributions differ, using term frequencies alone would require extensive manual pre-processing to select those particular terms (or n-grams) that are likely to discriminate well across categories, defeating the purpose of automatic content analysis.

We improve the discriminatory power of the input features by replacing each term with a *word embedding* technique, which takes each term and projects it into a lower dimensional vector space representation. The goal of this approach is to recover "word vectors" that encode similarities between otherwise lexically distinct words. Terms that convey similar concepts have "close" word vector representations. While the notion of word embeddings has been around since the 1980s (Rumelhart, Hinton, and Williams, 1986), recent advances in machine learning have permitted researchers to actually estimate models inferring latent word embedding representations from large text corpora within a reasonable timeframe (Mikolov, Chen, Corrado, and Dean, 2013). Additionally, the dimensions recovered from word vector models often convey substantively relevant concepts.[3] Word vector representations also provide a somewhat more theoretically grounded approach to pre-processing; although there are tweakable parameters, they do not require the long list of arbitrary steps such as deciding which stop words are uninformative. Our specific application of word embeddings uses the global word vector ("GloVe") model of Pennington, Socher, and Manning, 2014, which are 200-dimensional word vectors estimated

[3]Analogies between terms can also be represented by simple vector addition or subtraction. A common example in the literature is that evaluating the expression "King" - "Man" + "Woman" on the terms' respective word vectors yields a vector very close to that of the term "Queen" (Mikolov, Yih, and Zweig, 2013).

on a corpus of 2 billion Twitter posts with a vocabulary of about 1.2 million terms.

Using word vectors raises the question of how to create a document-feature matrix when each term in a document is no longer represented by a W-dimensional indicator vector (where W is the number of terms in the corpus vocabulary), but rather by a low-dimensional, continuous vector. While there are a variety of options, we find that taking the 10^{th}, 50^{th}, and 90^{th} quantiles of each word vector dimension yields a document-feature vector sufficiently rich for analysis, with other choices not mattering much.[4] Therefore, for any given corpus, we obtain a document-feature matrix F consisting of $W = 200 \times 3 = 600$ unique features observed for each document.

4.2 Improving feature discrimination

While different numerical representations capture different elements in the text, the goal of these transformations is to find a set of features that discriminate among the categories. Each category, in principle, should be associated with a separate underlying semantic concept. Additional dimensionality reduction may therefore help generate even more informative features. While it is possible to do so using techniques such as principal component analysis, these techniques are blind to the types of dimension reduction that would improve estimation, such as category membership. While dimension reduction methods like these have been developed for improving classifier performance (e.g., Brunzell and Eriksson, 2000; Vincent et al., 2010), no existing text-to-numbers algorithm we are aware of has been designed specifically for direct estimation of category proportions. We now outline the first such algorithm.

The intuition behind our approach is that even if individual features have poor discriminatory power, combinations of features may yield highly discriminatory features if those features exhibit meaningful correlations. Similarly, individual words convey some sense of the concepts a category represents, but sentences or paragraphs convey the concepts far better. Additionally, we thus directly optimize an objective function with respect to the discriminatory power across categories, which are observed and so should be used, while

[4] The choice of transformation here is akin to choosing a weighting scheme for individual terms when using raw terms as features. For example, a document-feature matrix consisting of raw counts of terms is implicitly taking a sum across all of the W-dimensional indicator vectors that represent each term.

reducing dimensions.

Methods We begin with the $N \times W$ document-feature matrix F defined in Section 4.1. In our case, we use 200 dimensional word vectors summarized by three points each, yielding $W = 600$, though in principle this method can be used for any document-feature matrix. Our goal in this section is to project this matrix to a lower-dimensional $N \times W'$ document-feature matrix $\underline{F} = F\Gamma$ where Γ is a $W \times W'$ matrix of transformation weights and $W' \ll W$. Once we obtain the lower-dimensional feature matrix $\underline{F}$, we can take conditional expectations of the features given the category labels to generate the readme regression matrix as before. We denote the regression matrix obtained from $\underline{F}$ as $\underline{X}$, in parallel to F and X. This new transformation can be thought of as a feed-forward neural network, where the input layer F feeds into a hidden layer $\underline{F}$, which produces an output $\underline{X}$.

To define optimal values of Γ we define an objective function that reflects two intuitions from Section 4. First, we want features with different conditional means across categories, which we call *category distinctiveness* (CD) and we want the rows of our regression matrix $\underline{X}$ to be not highly correlated with one another, which we call *feature distinctiveness* (FD). We define these criteria in their general form as

$$\text{CD}(\Gamma) \propto \sum_{c<c'} \sum_{w=1}^{W'} \left| \underline{X}^{\text{L}}_{wc} - \underline{X}^{\text{L}}_{wc'} \right|.$$

and

$$\text{FD}(\Gamma) \propto \sum_{c<c'} \sum_{w'<w} \left| \left| \underline{X}^{\text{L}}_{wc} - \underline{X}^{\text{L}}_{wc'} \right| - \left| \underline{X}^{\text{L}}_{w'c} - \underline{X}^{\text{L}}_{w'c'} \right| \right|.$$

where the inequalities in the summations prevent double-counting. We then choose Γ by optimizing the combination of these two criteria:

$$\Gamma^* = \underset{\Gamma \in \mathbb{R}^{W \times W'}}{\arg\max} \;\; \lambda \cdot \text{CD}(\Gamma) + (1 - \lambda) \cdot \text{FD}(\Gamma)$$

for some $\lambda \in [0, 1]$. In our experiments, we weight the two forms of distinctiveness equally ($\lambda = 0.5$), although this could be optimized further through a modified form of cross-validation or other means for some potential additional performance improvement.

The space of potential weight matrices is large, with many local minima, only some of which have good out-of-sample performance due to over-fitting. We thus take two steps to choose well.

First, we follow a principled approach by constraining the types of features that may be included in F. For example, if we were to generate $\underset{\sim}{X}^L$ in the labeled set by taking into account information about the resulting S in the unlabeled set, we could generate essentially any estimated unlabeled set category proportion $\hat{\pi}^U$ because we would be choosing both the left and right side of the readme regression. We therefore "tie our hands" by only including information from the labeled set in our optimization routine for Γ, which is reflected in the choice of objective function. As it turns out, tying one's hands is a more general statistical principle than our specific application. It has been appealed to implicitly or informally in many applications in the statistical literature, but never to our knowledge stated formally. We do so here:

Tied Hands Principle (THP). *Let t denote a function that transforms data objects A and B, into $A^* = t(A, Z)$ and $B^* = t(B, Z)$, by matching or weighting data subsets (rows), or transforming or selecting features (columns), where Z denotes exogenous information such that $p(A, B|Z) = p(A, B)$. Denote as $T_{A,Z}$ the set of all functions of A and Z (but not B) and $T_{B,Z}$ the set of all functions of B and Z (but not A). Then define $g(j|k)$ as a function to choose an element from set j using information in k. Consider statistical procedures that are functions of both A^* and B^*. Then, for $D = A$ or $D = B$, THP requires that the transformation function t be chosen such that $t = g(T_{D,Z}|D, Z)$ but not $t = g(T_{D,Z}|A, B, Z)$.*

The special case of the THP for readme2 prohibits finding Γ^* by minimizing $f(\Gamma, L, U)$. A different special case of the THP is commonly invoked in causal inference, where matching of treated and control observations is performed without being allowed to take into account the response variable: the observation weights are calculated explicitly without taking into account the outcome variable at all, or only in one treatment regime. Another special case in causal inference allows one, in prospective designs, to select observations conditional on the explanatory variables, but not the outcome variable or, in retro-

spective (case-control) designs, based on the outcome variable but not on the explanatory variables.

Second, while following THP, we incorporate techniques to reduce overfitting in our approach to estimating the weights. Specifically, we use stochastic gradient descent (SGD) with momentum (Kingma and Ba, 2015), a commonly-used optimization technique for fitting models by iteratively updating the model parameters based on the gradient of the objective function until convergence. SGD differs from the usual gradient descent in that it calculates the gradient update at each iteration using a randomly sampled single observation or (in "batch gradient descent") a subset of observations, which also makes it work well for large data sets. SGD is commonly used for fitting neural networks and its implementation in TensorFlow (Abadi et al., 2015), a popular software package for fitting large scale machine learning models, includes automatic differentiation to extract the gradients of the objective function. This is useful in our case since the derivative of the objective function is not easily obtained analytically.

Even when the data sets are relatively small, SGD has attractive regularizing properties, yielding solutions with good out-of-sample performance (Zhang et al., 2016). We augment this "implicit" regularization with a number of explicit techniques aimed to minimize overfitting. We incorporate "dropout" by randomly dropping some input features from contributing to some of the output features (essentially forcing their Γ weights to 0 and re-weighting other components of Γ to preserve the expected value of the resulting quantities) (Srivastava et al., 2014). This ensures some degree of sparsity in our overall Γ matrix. We also normalize the resulting features $\underline{F}$ to have mean zero and variance 1 and incorporate a confidence penalty as in (Pereyra et al., 2017). We also impose a maximum limit on the extent to which any single category or feature distinctiveness term contributes to the objective function (a process which is similar to gradient clipping). In our evaluations of the method, we set the number of final features, W', to 20 . Our experiments indicate that performance does not greatly depend on this parameter as long as it is much smaller than the number of input features W.

Illustration For our application, it is crucial that our objective function incorporates both category and feature distinctiveness. Optimizing Γ for category distinctiveness alone would lead to high variance through high collinearity in $\underset{\sim}{X}$, and optimizing Γ for feature distinctiveness alone would lead to higher bias and low category distinctiveness. Optimizing both together, as we do, reduces the overall error rate.

We illustrate this point with an analysis of data comprised of 1,426 emails drawn from the broader Enron Corporation email corpus made public during the Federal Energy Regulatory Commission's investigation into the firm's bankruptcy (the data and codebook are available at `j.mp/enronData` and `j.mp/EnronCodeBK`). These emails, also analyzed in Hopkins and King (2010) and below in Section 5, were hand coded by researchers into five broad topics: company business, personal communications, logistics arrangements, employment arrangements, and document editing.

For expository clarity, we set $W' = 2$ and choose Γ by first maximizing the category distinctiveness metric alone. We offer a scatterplot of the resulting projections, $\underset{\sim}{F}$, in the left panel of Figure 4, with different colors and symbols to represent the five categories. This panel reveals that these features do indeed discriminate between the categories (which can be seen by separation between the different colored symbols). However, as is also apparent, the two dimensions are highly correlated which, as in linear regression, would lead to higher variance estimates. In linear regression, given a fixed sample size, collinearity is an immutable fact of the fixed data set and specification; in contrast, in our application operating in the space of the features that we construct rather than the data we are given, we can change the projections, the space in which the regression is operating, and therefore the level of collinearity. As a second illustration, we again set $W' = 2$ but now optimize Γ by maximizing only feature distinctiveness. In this case, as can be seen in the middle panel of Figure 4, the columns of $\underset{\sim}{X}^{\mathrm{L}}$ are uncorrelated but unfortunately do not discriminate between categories well (as can be seen by the points with different colors and symbols overlapping).

Thus, we implement our metric, optimizing the sum of both category and feature distinctiveness, which we present in the right panel of Figure 4. This result is well calibrated

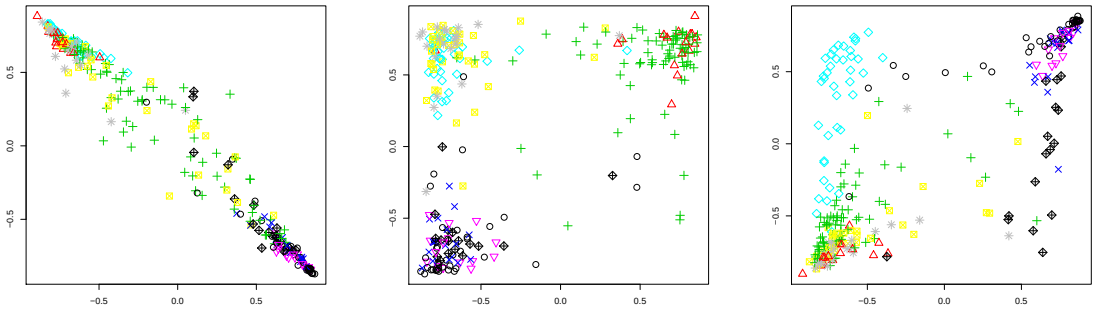

Figure 4: Optimizing Γ by category distinctiveness only (left panel), feature distinctivenss only (middle), and both (right panel). Each point is a document, with categories coded with a unique color and symbol. The axes (or components) are columns of tbe constructed $\phi(F \times \Gamma)$.

for estimating category proportions: The dimensions are discriminatory and thus bias reducing, which can be seen by the color separation, but still uncorrelated, and thus variance reducing. After matching on these features and performing the least squares regression in these Enron data, we find the sum of the absolute errors (SAE) in estimating π^{U} of 0.28, compared to 0.50 for the original readme, a substantial improvement.

4.3 Overcoming semantic change and proportion divergence

Given the results of Section 4.2, we could estimate π^{U} by applying the readme regression $(\underline{X}^{\mathrm{L}\prime}\underline{X}^{\mathrm{L}})^{-1}\underline{X}^{\mathrm{L}\prime}\underline{S}^{\mathrm{U}}$, with $\underline{X}^{\mathrm{L}}$ constructed from our newly optimized $\underline{F}$ matrix. However, we have found a way to further alter the space F (and therefore $\underline{X}^{\mathrm{L}}$), in a manner that reduces proportion divergence and any biasing effect of the vanishing discourse component of semantic change. The resulting method then depends less on the veracity of Assumption 2.

To do this, we borrow the idea of matching from the causal inference literature, and attempt to reduce the "imbalance" between the labeled and unlabeled sets on the observed feature distributions (Ho, Imai, King, and Stuart, 2007; Iacus, King, and Porro, 2012). In causal inference, we "prune" control observations from the analysis that have covariate values far from treated observations. In our case, we fix the unlabeled set and selectively prune the labeled set to remove documents with covariate profiles far from those in the unlabeled set. This enables us to remove much of the biasing effect of vanishing discourse

and to simultaneously reduce proportion divergence. We thus use the labeled set to construct a matched (sub)set, $\mathcal{M}$, that more closely resembles the unlabeled set. Note that we do this without any information from the *category* labels (since they are unobserved in the unlabeled set). Rather, we are removing observations from the labeled set that are so far from the observations seen in the unlabeled set that it is likely the case that they come from an entirely different data-generating process.

If the text of the documents are meaningful, exact full text matching in this way will eliminate all error since it will mean that we have a hand code for each unlabeled document. In practice, we take each document in the unlabeled set, identify the three nearest neighbors on $\underset{\sim}{F}$ in the labeled set (defined in the Euclidean space). Any labeled documents not matched by these rules are pruned and not used further. This act of pruning is what makes matching work in causal inference and, for our problem, reduces both semantic change and proportion divergence. We then recompute $\underset{\sim}{F}$ and the matched $\underset{\sim}{X}^{\mathrm{L}}$, which we denote $\underset{\sim}{X}^{\mathrm{L}\mathcal{M}}$, and apply the readme regression.

This pruning to the matched set change means that the assumption in Equation 2 needs only to hold in the matched subset of the labeled set rather than for the entire labeled set:

$$E[X^{\mathrm{L}\mathcal{M}}] = X^{\mathrm{U}}, \qquad \lim_{N^{\mathrm{L}}\to\infty} X^{\mathrm{L}\mathcal{M}} = X^{\mathrm{U}}. \tag{8}$$

Empirically, we find that matching indeed has the desired effects: in the 73 real-world data sets we introduce in Section 5, matching alone reduces the divergence between $\underset{\sim}{X}^{\mathrm{L}}$ and the true, unobserved $\underset{\sim}{X}^{\mathrm{U}}$ in 99.6% of cases and on average by 19.8%. Proportion divergence, which is not observed in real applications but which we can measure because we have coded unlabeled sets for evaluation, is reduced in 83.2% of data sets, on average by 25%. We now turn to the details of these data, and the consequence of using all parts of our new methods, for mean square error in the quantities of interest.

5 Evaluation

Design We performed 18 large-scale evaluations of our methodology, each following a different protocol for allocating documents to membership in the labeled and unlabeled sets. For each design protocol, we estimate readme2 and 32 alternative statistical methods

that can be used to estimate category proportions (including readme). Each method is analyzed on 19,710 ($= 73 \times 15 \times 18$) simulated data sets because we have 73 corpora, 15 iterations per corpora per design, and 18 designs. The total number of method-design observations is therefore $630,720$ ($= 19,710 \times 32$).

The 32 alternative methods of estimating category proportions are of five types. The first four types comprise six classifiers each run within each of the four possible combinations of (a) a discrete or continuous feature space and (b) a classification of whole documents and counting or averaging continuous probability estimates to yield estimates of the category proportions. The six classifiers include support vector machines, random forests, Naive Bayes, and L1- and L2-regularized multinomial regression (using standard settings and described in James, Witten, Hastie, and Tibshirani, 2013), and an ensemble of these classifiers based on an average of classifiers within each of the two cells of (b). The fifth type of alternative method includes 8 methods tuned for quantification. Among these, only readme is designed for more than two categories. We adapt the remaining 7 — Friedman, Adjusted Counts, HDX, Median Sweep, Mixture HPMF, Mixture L1, and Mixture L2 (each detailed in Firat 2016) — to multiple categories via estimation of repeated dichtomizations of the set of categories.

Each of the 19,710 data sets we analyze, constructed as a subset of one of 73 corpora, has a labeled out-of-sample test set that plays the role of the unlabeled set, except that we are able to use its labels after estimation to evaluate performance. The 73 corpora include three used in Hopkins and King (2010): The Enron email data set described in Section 4.2; a set of 462 newspaper editorials about immigration (with 3,618 word stems and 5 categories); and a set with 1,938 blog posts about candidate Hillary Clinton from the 2008 presidential election (with 3,623 word stems and 7 categories). We also include 11,855 sentences (with 5 categories and 3,618 word stems) from the Stanford Sentiment Treebank (Socher et al., 2013). Finally, we include 69 separate social media data sets (most from Twitter and a few from diverse blogs and Facebook posts), each created by a different political candidate, private company, nonprofit, or government agency for their own purposes, covering different time frames and categorization schemes (see Firat, 2016);

these data cover 150–4,200 word stems, 3–12 categories, and 700–4,000 tweets. (All data are in our replication data set, except that for privacy reasons the raw text of the 69 has been coded as numbers.)

Nearly all documents in the 73 corpora are labeled with a time stamp. For the empirical design, we randomly select a time point and pick the previous 300 documents as the labeled set and the next 300 documents as the out-of-sample evaluation set (wrapping in time if necessary). For each corpora, we repeat this process 50 times. This procedure keeps the evaluation highly realistic while also ensuring that we have many types of data sets with variation in proportion divergence, textual discrimination, and semantic change. The joint distribution of these quantities is extremely important in determining the overall error dynamics, so accurately simulating this distribution is of the utmost importance in this exercise. Although we argue that the empirical design is particularly realistic, we replicate our analysis across 18 designs which are described in Appendix B and which make different assumptions about the joint distribution just discussed.

Prior evaluative approaches in the literature have almost always used a single simulation design, as compared to our 18, and only a few data sets, compared to our 73. The resulting 19,710 empirical evaluations in our replication data and code thus greatly increases the rigor which future scholars can bring to bear on new methods developed to improve on those proposed here.

Results We present results across our numerous evaluations in three ways.

First, Figure 5 compares the performance of readme2 to the 32 alternative methods across all 18 designs. For each method, we compute the proportion of data sets with higher error than readme2 vertically by the proportion divergence in quantiles horizontally. Our new approach outperforms the best classifier (in these data, a support vector machine (SVM) model run in the continuous feature space) in 94.5%of corpora. Many of the 32 methods are outperformed by readme2 in 100% of the cases, as indicated by appearing at the top of the graph. Relative performance remains excellent across the different levels of category proportion divergence between labeled and unlabeled sets. The new method's relative performance improves when proportion divergence is high (at the

right, with more substantial changes between labeled and unlabeled sets), which makes sense since ours is the only approach to directly address semantic change. The different types of methods (represented as lines) follow three basic patterns in relative performance: (a) classifiers with averaged probabilities (in black and green) have higher SAE relative to readme2 as divergence increases, due to their assumption that test and training sets are drawn from the same distribution; (b) quantification methods (in light blue) approach readme2's performance only with high levels of divergence, since they are designed for this situation; and (c) the remaining methods perform relatively poorly overall regardless of proportion divergence.

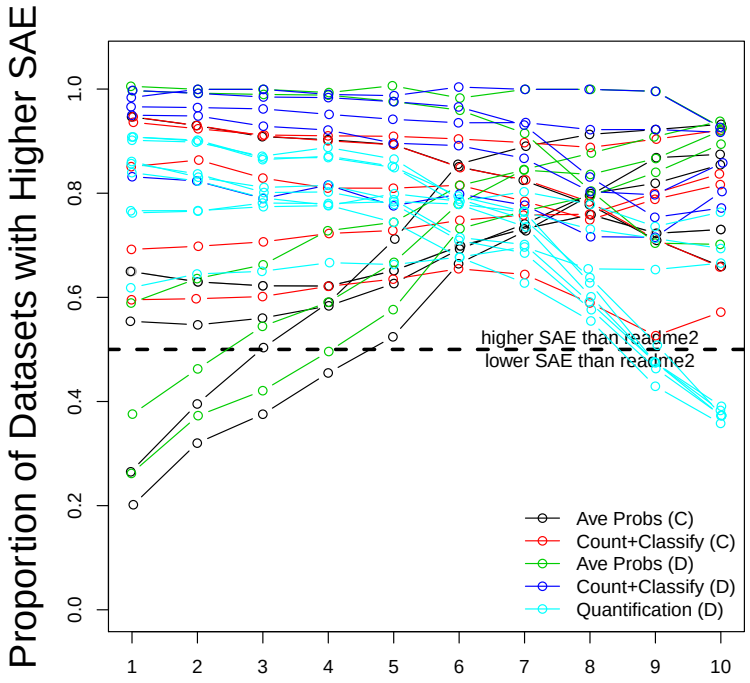

Figure 5: The proportion of corpora with higher error than our approach (vertically) by quantile in proportion divergence (horizontally), with 32 different methods color coded by type (described in the text, with feature space "D" for discrete and "C"for continuous).

Second, we provide a more detailed comparison of the performance of readme to readme2, the primary goal of this paper. In the empirical design, which we argue is particularly important in practice, we find an 35.3% average corpus-wide improvement over readme, which in terms of SAE is a substantial 9.4 percentage points. Figure 6

plots estimation error (vertically) for readme compared to our new approach (ordered horizontally by size of the improvement). The length of each arrow represents the average improvement over subsets of each of the 73 corpora, with one arrow for each. In all cases, the arrows face downward, meaning that in every corpus our new method outperforms readme on average. Our new approach performs better than all three of the data sets used in Hopkins and King (2010), and also the Stanford Sentiment data set (the colored arrows).

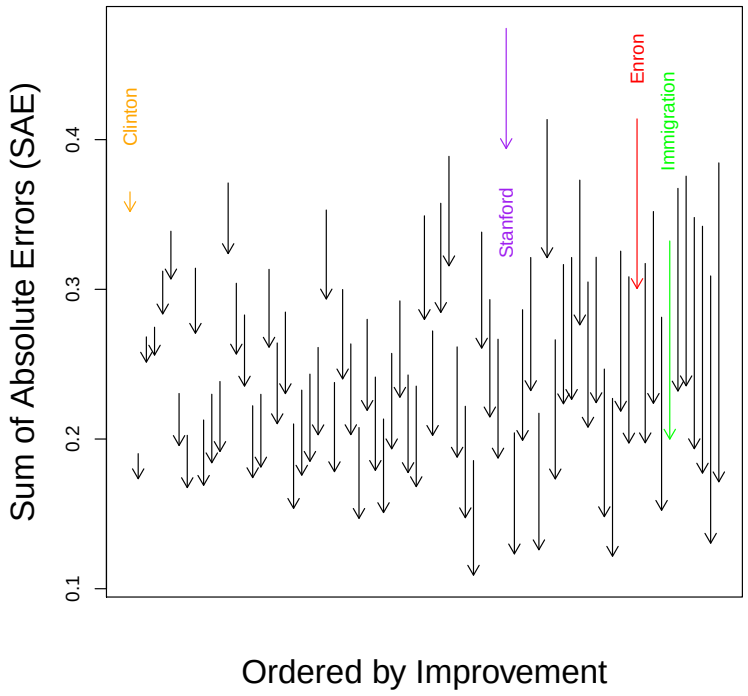

Figure 6: Average Error Reduction: readme to readme2, in analyses of 50 subsets of each of the 73 data sets. The length of each arrow indicates the change in performance, with downward arrows indicating how SAE is reduced by our new methodology. Colored arrows refer to the four publicly available data sets, three of which were used in Hopkins and King (2010).

Finally, we show that our results are robust across our 18 diverse simulation designs (described in Appendix B). The left panel of Figure 7 compares average performance over simulations and reveals that and readme2 outperforms readme for every simulation design one (as indicated by being above the dotted horizontal line). The empirical analysis, noted

in red, is the substantively most meaningful design described above. Then, the right panel of Figure 7 illustrates how, across the 18 simulation designs, readme2 outperforms not only readme, but all 32 alternative methods in a large fraction of cases. Readme2's average rank is 3.89, whereas the next best algorithm's average rank is 6.50. Median performance (indicated by the horizontal gray bar for each design) is always improved.

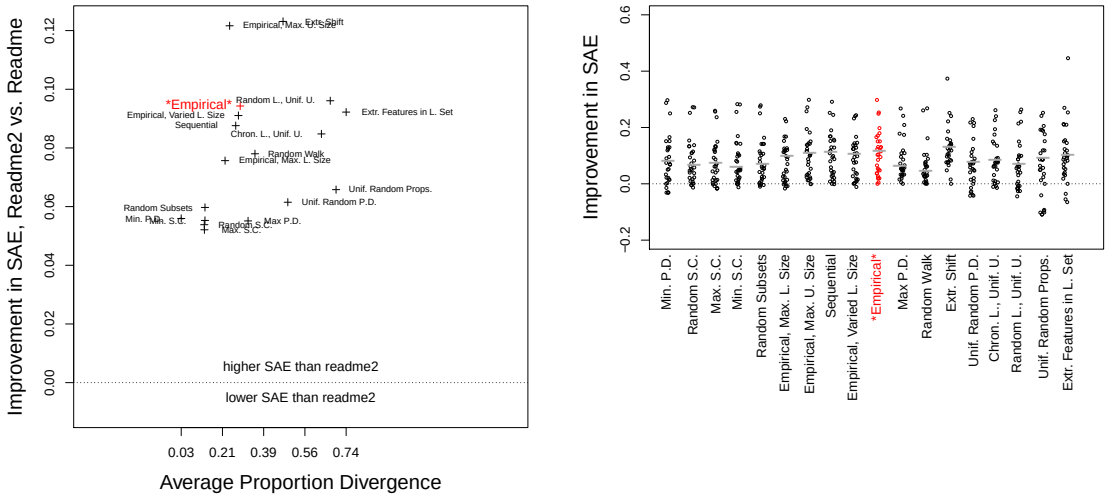

Figure 7: Average Error Reduction Across Simulation Designs, relative to readme (left panel) and 32 alternative methods (right panel). The simulation marked Empirical in both is the one described in the text; for the others, see Appendix B. Items above the dotted horizontal line in both panels indicate that readme2 reduced SAE compared to a competing method. The gray horizontal line for each set of simulations in the right panel is the median. "P.D." stands for "Proportion Divergence"; "S.C." stands for "Semantic Change"; "L." stands for "Labeled"; "U" stands for "Unlabeled."

In sum, our new methodology would seem to be preferably to readme and other existing methods across a wide array of corpora, data sets, and evaluation protocols. It is also worth noting that readme2 is computationally fast due to its use of batch sampling and efficient symbolic math libraries. Estimation on a dataset with a labeled set size of 1,000 and an unlabeled set size of 500, with 5 categories and 600 raw features, takes about 11.5 seconds on a CPU with a 2.7 GHz processor and 8 GB of memory (while estimation via SVM, for example, takes 10.5 seconds and via lasso-regularized regression takes 7.3 seconds).

6 What Can Go Wrong?

Our results indicate that the methods introduced here are clear improvements over readme and other approaches, and this conclusion is robust in a wide variety of circumstances, category types, and corpora. We focus in this section on the three situations we have found where our approach may not help and further research may be productive.

First, when we use matching in continuous space, we generally reduce proportion divergence and the effects of vanishing discourse. However, emerging discourse can not only cause bias (in any method), but this bias can sometimes be induced by the analyst in the process of dealing with vanishing discourse. In addition, although readme2 is the only method that has been proposed to reduce the effects of vanishing discourse, the method is of no help if all the relevant discourse vanishes within a category. This is akin to a violation of the common support assumption in matching methods used in causal inference and so must rely on risky extrapolations. Unlike with classifiers, our methodology does not need to assume that the labeled and unlabeled sets are drawn from the same distribution, but we do require that the distributions have some overlap. If one suspects that meaning or language is changing dramatically, an easy fix is to code additional observations from later points in time.

Second, if the original feature space is highly sparse (as in a regression with a large number of irrelevant covariates), then our optimization algorithm may have difficulty arriving at a stable solution for Γ. This can happen with highly uninformative text, categories with labels that may be more meaningful to investigators than the authors of the text, or error-ridden hand coding. If the word vectors used to generate the raw features were trained on an inappropriate corpus, performance would also be expected to deteriorate, as the relationship between the text and numbers would be more tenuous. Our word vectors are from Twitter and so we recommend swapping these out with another if the text being analyzed differs substantially from tweets. Pre-trained word vectors now exist for many languages as well.

Finally, we emphasize that our approach relies on meaningful text in each document, conceptually coherent and mutually exclusive and exhaustive categories, and a labeling

effort that validly and reliably codes documents into the right categories. These may seem like obvious criteria, but they always constitute the most important steps in any automated text analysis method, including ours. In our experience most of the effort in getting an analysis right involves, or should involve, these preliminary steps.

7 Concluding Remarks

We improve on a popular method of estimating category proportions (King and Lu, 2008; Hopkins and King, 2010), a task of central interest to social scientists among others. We do this without having to tune or even use the often model dependent methods of individual classification developed in other fields for different quantities of interest. We prove properties and provide intuition about readme and then build our alternative. We have tested our analysis in 73 separate data sets, 19,710 data subsets, and 18 evaluation protocols, with encouraging results. Overall, our approach weakens the key assumptions of readme while creating new, more meaningful numerical representations of each of the documents specifically tuned to reduce the mean square error of multi-category, nonparametric quantification.

We can identify several ways of building on our work to further improve performance. These include methods for optimizing the raw continuous textual representations used in readme2. In this analysis, we use document-level summaries of word vectors for the raw features, but there is no quantitative principle implying that this choice is optimal and so could be improved. Indeed, our results suggest that the quantitative features used in readme greatly influence the performance of the estimator. It is natural, then, to consider continuous document-level representations directly from the labeled (and unlabeled) sets, or possibly using category-wise information from the labeled set or with smoothing toward word vectors created from giant corpora such as we use from Twitter. We could also optimize over λ, rather than setting it to 0.5 as we do now, among other related small changes. With these additions, the estimation process could be more fully optimized for quantification. Finally, further work could explore more systematically the application of these ideas to other non-parametric methods.

Appendix A Bias on the Simplex in Two Categories

Proof of Proposition 1

Start with the least-squares minimization problem

$$\widehat{\pi_1^{\mathrm{U}}} = \arg\min_{\pi_1^U} \sum_{w=1}^{W} \left(S_w^{\mathrm{U}} - \widehat{S_w^{\mathrm{U}}} \right)^2.$$

Write $\widehat{S_w^{\mathrm{U}}}$ in terms of X^{L} and π_1^{U}

$$\widehat{\pi_1^{\mathrm{U}}} = \arg\min_{\pi_1^U} \sum_{w=1}^{W} \left(S_w^{\mathrm{U}} - X_{w2}^{\mathrm{L}} - \left(X_{w1}^{\mathrm{L}} - X_{w2}^{\mathrm{L}} \right) \pi_1^{\mathrm{U}} \right)^2.$$

Take the derivative and set equal to 0

$$\begin{aligned}
0 &= \frac{\partial}{\partial \pi_1^{\mathrm{U}}} \sum_{w=1}^{W} \left(S_w^{\mathrm{U}} - X_{w2}^{\mathrm{L}} - \left(X_{w1}^{\mathrm{L}} - X_{w2}^{\mathrm{L}} \right) \pi_1^{\mathrm{U}} \right)^2 \\
&= \sum_{w=1}^{W} \left(S_w^{\mathrm{U}} - X_{w2}^{\mathrm{L}} \right) \left(X_{w1}^{\mathrm{L}} - X_{w2}^{\mathrm{L}} \right) - \left(X_{w1}^{\mathrm{L}} - X_{w2}^{\mathrm{L}} \right)^2 \pi_1^{\mathrm{U}} \\
\sum_{w=1}^{W} \left(S_w^{\mathrm{U}} - X_{w2}^{\mathrm{L}} \right) \left(X_{w1}^{\mathrm{L}} - X_{w2}^{\mathrm{L}} \right) &= \sum_{w=1}^{W} \left(X_{w1}^{\mathrm{L}} - X_{w2}^{\mathrm{L}} \right)^2 \pi_1^{\mathrm{U}} \\
\frac{\sum_{w=1}^{W} \left(S_w^{\mathrm{U}} - X_{w2}^{\mathrm{L}} \right) \left(X_{w1}^{\mathrm{L}} - X_{w2}^{\mathrm{L}} \right)}{\sum_{w=1}^{W} \left(X_{w1}^{\mathrm{L}} - X_{w2}^{\mathrm{L}} \right)^2} &= \pi_1^{\mathrm{U}}.
\end{aligned}$$

Since the expression being optimized is quadratic, this is a global optimum. Therefore the readme estimator in two categories has the closed-form expression

$$\widehat{\pi_1^{\mathrm{U}}} = \frac{\sum_{w=1}^{W} \left(X_{w1}^{\mathrm{L}} - X_{w2}^{\mathrm{L}} \right) \left(S_w^{\mathrm{U}} - X_{w2}^{\mathrm{L}} \right)}{\sum_{w=1}^{W} \left(X_{w1}^{\mathrm{L}} - X_{w2}^{\mathrm{L}} \right)^2}.$$

Proof of Proposition 2

Start with the expression for $\widehat{\pi_1^{\mathrm{U}}}$.

$$\widehat{\pi_1^{\mathrm{U}}} = \frac{\sum_{w=1}^{W} \left(X_{w1}^{\mathrm{L}} - X_{w2}^{\mathrm{L}} \right) \left(S_w^{\mathrm{U}} - X_{w2}^{\mathrm{L}} \right)}{\sum_{w=1}^{W} \left(X_{w1}^{\mathrm{L}} - X_{w2}^{\mathrm{L}} \right)^2}.$$

Write X_{wc}^{L} in terms of X_{wc}^{U} and ϵ_{wc}.

$$\widehat{\pi_1^{\mathrm{U}}} = \frac{\sum_{w=1}^{W} \left(X_{w1}^{\mathrm{U}} - X_{w2}^{\mathrm{U}} + \epsilon_{w1} - \epsilon_{w2}\right)\left(S_w^{\mathrm{U}} - X_{w2}^{\mathrm{U}} - \epsilon_{w2}\right)}{\sum_{w=1}^{W} \left(X_{w1}^{\mathrm{U}} - X_{w2}^{\mathrm{U}} + \epsilon_{w1} - \epsilon_{w2}\right)^2}.$$

Substitute the accounting identity for S_w^{U}

$$\widehat{\pi_1^{\mathrm{U}}} = \frac{\sum_{w=1}^{W} \left(X_{w1}^{\mathrm{U}} - X_{w2}^{\mathrm{U}} + \epsilon_{w1} - \epsilon_{w2}\right)\left((X_{w1}^{\mathrm{U}} - X_{w2}^{\mathrm{U}})\pi_1^{\mathrm{U}} - \epsilon_{w2}\right)}{\sum_{w=1}^{W} \left(X_{w1}^{\mathrm{U}} - X_{w2}^{\mathrm{U}} + \epsilon_{w1} - \epsilon_{w2}\right)^2}.$$

Expanding the numerator

$$\widehat{\pi_1^{\mathrm{U}}} = \frac{\sum_{w=1}^{W} \left(X_{w1}^{\mathrm{U}} - X_{w2}^{\mathrm{U}} + \epsilon_{w1} - \epsilon_{w2}\right)\left(X_{w1}^{\mathrm{U}} - X_{w2}^{\mathrm{U}}\right)}{\sum_{w=1}^{W} \left(X_{w1}^{\mathrm{U}} - X_{w2}^{\mathrm{U}} + \epsilon_{w1} - \epsilon_{w2}\right)^2}\pi_1^{\mathrm{U}} - \frac{\sum_{w=1}^{W} \left(X_{w1}^{\mathrm{U}} - X_{w2}^{\mathrm{U}} + \epsilon_{w1} - \epsilon_{w2}\right)\left(\epsilon_{w2}\right)}{\sum_{w=1}^{W} \left(X_{w1}^{\mathrm{U}} - X_{w2}^{\mathrm{U}} + \epsilon_{w1} - \epsilon_{w2}\right)^2}.$$

Taking the expectation, we find

$$E\left[\widehat{\pi_1^{\mathrm{U}}}\right] = E\left[\frac{\sum_{w=1}^{W} \left(X_{w1}^{\mathrm{U}} - X_{w2}^{\mathrm{U}} + \epsilon_{w1} - \epsilon_{w2}\right)\left(X_{w1}^{\mathrm{U}} - X_{w2}^{\mathrm{U}}\right)}{\sum_{w=1}^{W} \left(X_{w1}^{\mathrm{U}} - X_{w2}^{\mathrm{U}} + \epsilon_{w1} - \epsilon_{w2}\right)^2}\right]\pi_1^{\mathrm{U}} - E\left[\frac{\sum_{w=1}^{W} \left(X_{w1}^{\mathrm{U}} - X_{w2}^{\mathrm{U}} + \epsilon_{w1} - \epsilon_{w2}\right)\left(\epsilon_{w2}\right)}{\sum_{w=1}^{W} \left(X_{w1}^{\mathrm{U}} - X_{w2}^{\mathrm{U}} + \epsilon_{w1} - \epsilon_{w2}\right)^2}\right].$$

Proof of Proposition 3

Using the first-order Taylor approximation $E\left[\frac{X}{Y}\right] \approx \frac{E[X]}{E[Y]}$, we have

$$\begin{aligned}
E\left[\widehat{\pi_1^{\mathrm{U}}}\right] &\approx \frac{E\left[\sum_{w=1}^{W} \left(X_{w1}^{\mathrm{U}} - X_{w2}^{\mathrm{U}} + \epsilon_{w1} - \epsilon_{w2}\right)\left(X_{w1}^{\mathrm{U}} - X_{w2}^{\mathrm{U}}\right)\right]}{E\left[\sum_{w=1}^{W} \left(X_{w1}^{\mathrm{U}} - X_{w2}^{\mathrm{U}} + \epsilon_{w1} - \epsilon_{w2}\right)^2\right]}\pi_1^{\mathrm{U}} - \frac{E\left[\sum_{w=1}^{W} \left(X_{w1}^{\mathrm{U}} - X_{w2}^{\mathrm{U}} + \epsilon_{w1} - \epsilon_{w2}\right)\left(\epsilon_{w2}\right)\right]}{E\left[\sum_{w=1}^{W} \left(X_{w1}^{\mathrm{U}} - X_{w2}^{\mathrm{U}} + \epsilon_{w1} - \epsilon_{w2}\right)^2\right]} \\
&= \frac{\sum_{w=1}^{W} \left(X_{w1}^{\mathrm{U}} - X_{w2}^{\mathrm{U}}\right)^2}{\sum_{w=1}^{W} (X_{w1}^{\mathrm{U}} - X_{w2}^{\mathrm{U}})^2 + E[(\epsilon_{w1} - \epsilon_{w2})^2]}\pi_1^{\mathrm{U}} - \frac{\sum_{w=1}^{W} E[\epsilon_{w1}\epsilon_{w2}] - E[(\epsilon_{w2})^2]}{\sum_{w=1}^{W} (X_{w1}^{\mathrm{U}} - X_{w2}^{\mathrm{U}})^2 + E[(\epsilon_{w1} - \epsilon_{w2})^2]} \\
&= \frac{\sum_{w=1}^{W} \left(X_{w1}^{\mathrm{U}} - X_{w2}^{\mathrm{U}}\right)^2 \pi_1^{\mathrm{U}} - E[\epsilon_{w1}\epsilon_{w2}] + E[(\epsilon_{w2})^2]}{\sum_{w=1}^{W} (X_{w1}^{\mathrm{U}} - X_{w2}^{\mathrm{U}})^2 + E[(\epsilon_{w1} - \epsilon_{w2})^2]} \\
&= \frac{\sum_{w=1}^{W} \left(X_{w1}^{\mathrm{U}} - X_{w2}^{\mathrm{U}}\right)^2 \pi_1^{\mathrm{U}} + \mathrm{Var}(\epsilon_{w2}) - \mathrm{Cov}(\epsilon_{w1}, \epsilon_{w2})}{\sum_{w=1}^{W} (X_{w1}^{\mathrm{U}} - X_{w2}^{\mathrm{U}})^2 + \mathrm{Var}(\epsilon_{w1} - \epsilon_{w2})} \\
&= \frac{\sum_{w=1}^{W} \left(X_{w1}^{\mathrm{U}} - X_{w2}^{\mathrm{U}}\right)^2 \pi_1^{\mathrm{U}} + \mathrm{Var}(\epsilon_{w2}) - \mathrm{Cov}(\epsilon_{w1}, \epsilon_{w2})}{\sum_{w=1}^{W} (X_{w1}^{\mathrm{U}} - X_{w2}^{\mathrm{U}})^2 + \mathrm{Var}(\epsilon_{w1}) + \mathrm{Var}(\epsilon_{w2}) - 2\mathrm{Cov}(\epsilon_{w1}, \epsilon_{w2})},
\end{aligned}$$

where the last two lines follow from the definition of variance and the assumption that $E[\epsilon_{wc}] = 0$. Subtracting π_1^{U} to get the bias:

$$\mathrm{Bias}(\widehat{\pi_1^{\mathrm{U}}}) \approx \frac{\sum_{w=1}^{W} \left(X_{w1}^{\mathrm{U}} - X_{w2}^{\mathrm{U}}\right)^2 \pi_1^{\mathrm{U}} + \mathrm{Var}(\epsilon_{w2}) - \mathrm{Cov}(\epsilon_{w1}, \epsilon_{w2})}{\sum_{w=1}^{W} (X_{w1}^{\mathrm{U}} - X_{w2}^{\mathrm{U}})^2 + \mathrm{Var}(\epsilon_{w1}) + \mathrm{Var}(\epsilon_{w2}) - 2\mathrm{Cov}(\epsilon_{w1}, \epsilon_{w2})} - \pi_1^{\mathrm{U}}$$

$$
\begin{aligned}
&= \frac{\sum_{w=1}^{W} \text{Var}(\epsilon_{w2}) - \text{Cov}(\epsilon_{w1}, \epsilon_{w2}) - \text{Var}(\epsilon_{w1})\pi_1^{\text{U}} - \text{Var}(\epsilon_{w2})\pi_1^{\text{U}} + 2\text{Cov}(\epsilon_{w1}, \epsilon_{w2})\pi_1^{\text{U}}}{\sum_{w=1}^{W}(X_{w1}^{\text{U}} - X_{w2}^{\text{U}})^2 + \text{Var}(\epsilon_{w1}) + \text{Var}(\epsilon_{w2}) - 2\text{Cov}(\epsilon_{w1}, \epsilon_{w2})} \\
&= \frac{\sum_{w=1}^{W} \text{Var}(\epsilon_{w2})(1 - \pi_1^{\text{U}}) - \text{Var}(\epsilon_{w1})\pi_1^{\text{U}} - \text{Cov}(\epsilon_{w1}, \epsilon_{w2}) + 2\text{Cov}(\epsilon_{w1}, \epsilon_{w2})\pi_1^{\text{U}}}{\sum_{w=1}^{W}(X_{w1}^{\text{U}} - X_{w2}^{\text{U}})^2 + \text{Var}(\epsilon_{w1}) + \text{Var}(\epsilon_{w2}) - 2\text{Cov}(\epsilon_{w1}, \epsilon_{w2})} \\
&= \frac{\sum_{w=1}^{W} \text{Var}(\epsilon_{w2})(1 - \pi_1^{\text{U}}) - \text{Var}(\epsilon_{w1})\pi_1^{\text{U}} - \text{Cov}(\epsilon_{w1}, \epsilon_{w2})(1 - \pi_1^{\text{U}}) + \text{Cov}(\epsilon_{w1}, \epsilon_{w2})\pi_1^{\text{U}}}{\sum_{w=1}^{W}(X_{w1}^{\text{U}} - X_{w2}^{\text{U}})^2 + \text{Var}(\epsilon_{w1}) + \text{Var}(\epsilon_{w2}) - 2\text{Cov}(\epsilon_{w1}, \epsilon_{w2})} \\
&= \frac{\sum_{w=1}^{W} \left(\text{Var}(\epsilon_{w2}) - \text{Cov}(\epsilon_{w1}, \epsilon_{w2})\right)(1 - \pi_1^{\text{U}}) - \left(\text{Var}(\epsilon_{w1}) - \text{Cov}(\epsilon_{w1}, \epsilon_{w2})\right)\pi_1^{\text{U}}}{\sum_{w=1}^{W}(X_{w1}^{\text{U}} - X_{w2}^{\text{U}})^2 + \text{Var}(\epsilon_{w1}) + \text{Var}(\epsilon_{w2}) - 2\text{Cov}(\epsilon_{w1}, \epsilon_{w2})}.
\end{aligned}
$$

Proof of Proposition 4

Substituting in the known measurement error variances and assuming independence in measurement errors across categories yields:

$$
\text{Bias}(\widehat{\pi_1^{\text{U}}}) \approx \frac{\sum_{w=1}^{W} \left(\frac{\sigma_{w2}^2}{N_2^{\text{L}}}\right)(1 - \pi_1^{\text{U}}) - \left(\frac{\sigma_{w1}^2}{N_1^{\text{L}}}\right)\pi_1^{\text{U}}}{\sum_{w=1}^{W}(X_{w1}^{\text{U}} - X_{w2}^{\text{U}})^2 + \left(\frac{\sigma_{w2}^2}{N_2^{\text{L}}}\right) + \left(\frac{\sigma_{w1}^2}{N_1^{\text{L}}}\right)}
$$

Using the fact that $N_c^{\text{L}} = N^{\text{L}}\pi_c^{\text{L}}$

$$
\begin{aligned}
\text{Bias}(\widehat{\pi_1^{\text{U}}}) &\approx \frac{(1 - \pi_1^{\text{U}})\sum_{w=1}^{W} \left(\frac{\sigma_{w2}^2}{N^{\text{L}}\pi_2^{\text{L}}}\right) - \pi_1^{\text{U}}\sum_{w=1}^{W} \left(\frac{\sigma_{w1}^2}{N^{\text{L}}\pi_1^{\text{L}}}\right)}{\sum_{w=1}^{W}(X_{w1}^{\text{U}} - X_{w2}^{\text{U}})^2 + \left(\frac{\sigma_{w2}^2}{N^{\text{L}}\pi_2^{\text{L}}}\right) + \left(\frac{\sigma_{w1}^2}{N^{\text{L}}\pi_1^{\text{L}}}\right)} \\
&\approx \frac{\frac{1}{N^{\text{L}}}\left[\frac{(1-\pi_1^{\text{U}})}{(1-\pi_1^{\text{L}})}\sum_{w=1}^{W}\sigma_{w2}^2 - \frac{\pi_1^{\text{U}}}{\pi_1^{\text{L}}}\sum_{w=1}^{W}\sigma_{w1}^2\right]}{\sum_{w=1}^{W}(X_{w1}^{\text{U}} - X_{w2}^{\text{U}})^2 + \frac{1}{N^{L}(1-\pi_1^{\text{L}})}\sum_{w=1}^{W}\sigma_{w2}^2 + \frac{1}{N^{L}\pi_1^{\text{L}}}\sum_{w=1}^{W}\sigma_{w1}^2} \\
&\approx \frac{\frac{(1-\pi_1^{\text{U}})}{(1-\pi_1^{\text{L}})}\sum_{w=1}^{W}\sigma_{w2}^2 - \frac{\pi_1^{\text{U}}}{\pi_1^{\text{L}}}\sum_{w=1}^{W}\sigma_{w1}^2}{N^{\text{L}}\sum_{w=1}^{W}(X_{w1}^{\text{U}} - X_{w2}^{\text{U}})^2 + \frac{1}{(1-\pi_1^{\text{L}})}\sum_{w=1}^{W}\sigma_{w2}^2 + \frac{1}{\pi_1^{\text{L}}}\sum_{w=1}^{W}\sigma_{w1}^2}
\end{aligned}
$$

The denominator is strictly positive. Therefore, bias is minimized when the numerator is equal to 0. Solving for π_1^{U} in terms of π_1^{L} yields

$$
\begin{aligned}
0 &= \frac{(1 - \pi_1^{\text{U}})}{(1 - \pi_1^{\text{L}})}\sum_{w=1}^{W}\sigma_{w2}^2 - \frac{\pi_1^{\text{U}}}{\pi_1^{\text{L}}}\sum_{w=1}^{W}\sigma_{w1}^2 \\
\frac{\pi_1^{\text{U}}}{\pi_1^{\text{L}}}\sum_{w=1}^{W}\sigma_{w1}^2 &= \frac{(1 - \pi_1^{\text{U}})}{(1 - \pi_1^{\text{L}})}\sum_{w=1}^{W}\sigma_{w2}^2 \\
\pi_1^{\text{U}}(1 - \pi_1^{\text{L}})\sum_{w=1}^{W}\sigma_{w1}^2 &= (1 - \pi_1^{\text{U}})\pi_1^{\text{L}}\sum_{w=1}^{W}\sigma_{w2}^2
\end{aligned}
$$

$$\left(\pi_1^{\mathrm{U}} - \pi_1^{\mathrm{U}}\pi_1^{\mathrm{L}}\right)\sum_{w=1}^{W}\sigma_{w1}^2 = \left(\pi_1^{\mathrm{L}} - \pi_1^{\mathrm{U}}\pi_1^{\mathrm{L}}\right)\sum_{w=1}^{W}\sigma_{w2}^2$$

$$\pi_1^{\mathrm{U}}\sum_{w=1}^{W}\sigma_{w1}^2 = \pi_1^{\mathrm{L}}\sum_{w=1}^{W}\sigma_{w2}^2 + \pi_1^{\mathrm{U}}\pi_1^{\mathrm{L}}\left(\sum_{w=1}^{W}\sigma_{w1}^2 - \sigma_{w2}^2\right)$$

$$\frac{\pi_1^{\mathrm{U}}\sum_{w=1}^{W}\sigma_{w1}^2}{\sum_{w=1}^{W}\sigma_{w2}^2 + \pi_1^{\mathrm{U}}\left(\sum_{w=1}^{W}\sigma_{w1}^2 - \sigma_{w2}^2\right)} = \pi_1^{\mathrm{L}}$$

$$\frac{\pi_1^{\mathrm{U}}\sum_{w=1}^{W}\sigma_{w1}^2}{\pi_1^{\mathrm{U}}\sum_{w=1}^{W}\sigma_{w1}^2 + (1-\pi_1^{\mathrm{U}})\sum_{w=1}^{W}\sigma_{w2}^2} = \pi_1^{\mathrm{L}}$$

When the measurement error variances are generally constant across categories, the bias is zero when the labeled set proportions are equal to the unlabeled set proportions.

Appendix B Alternative Evaluation Designs

Each of the 18 evaluation designs summarized in Table 1 offers a different way of generating 19,710 data sets as subsets of the 73 corpora described in Section 5. Each data set is divided into a labeled set as well as a test set that serves the purpose of the unlabeled set during estimation, but can also be used for evaluation since all its document labels are observed.

Table 1: Alternative Evaluation Designs

Design Name	**Description**
Empirical	Sample consecutive chronological slices of the data to form labeled and unlabeled sets, wrapping when necessary (detailed in Section 5).
Empirical, Varied Labeled Set Size	Sample consecutive chronological slices of the data to form labeled and unlabeled sets, wrapping when necessary (detailed in Section 5). Randomly sample the labeled set size from $\{100, 300, 500, 1000\}$.
Empirical, Maximum Labeled Set Size	Sample a consecutive chronological slice of the data to form the labeled set. Use the remainder of the documents to form the unlabeled set.
Empirical, Maximum Unlabeled Set Size	Sample a consecutive chronological slice of the data to form the labeled set. Use the remaining 300 documents to form the unlabeled set.
Sequential	Sample a time point randomly. From the 300 documents preceding this date, form the labeled set. From the 300 documents following to this date, form the unlabeled set. Wrap when necessary.
Random Subsets	Sample documents randomly without replacement for labeled and unlabeled sets.

Table 1 . . . Continued

Design Name	**Description**
Min. Proportion Divergence	Sample documents randomly without replacement to form 10,000 candidate labeled and unlabeled sets. Select the pair which minimizes $\lvert\pi^{\mathrm{L}} - \pi^{\mathrm{U}}\rvert$ divergence.
Uniform Random Proportion Divergence	Draw random uniform on the interval $[0, 0.75]$, the target $\lvert\pi^{\mathrm{L}} - \pi^{\mathrm{U}}\rvert$ divergence. Draw 10,000 candidate π^{L} and π^{U} uniform from the simplex. Select the pair closest to the target.
Random Labeled Set, Uniformly Random $\Pr(D)$ in Unlabeled Set	Draw 300 labeled set documents at random from the set of candidates. Draw a target $\Pr(D)^U$ uniformly from the simplex and select candidate documents to achieve this target.
Max. Proportion Divergence	Sample documents randomly without replacement to form 10,000 candidate labeled and unlabeled sets. Select pair that maximizes $\lvert\pi^{\mathrm{L}} - \pi^{\mathrm{U}}\rvert$ divergence.
Min. Semantic Change	Sample documents randomly without replacement to form 10,000 candidate labeled and unlabeled sets. Select the pair that minimizes semantic change.
Uniform Random Semantic Change	Sample documents randomly without replacement to form 10,000 candidate labeled and unlabeled sets. Select a uniform random target amount of semantic change. Select the pair closest to the target.
Max. Semantic Change	Sample documents randomly without replacement to form 10,000 candidate labeled and unlabeled sets. Select the pair which maximizes semantic change.
Random Walk	Draw π^{L} from a uniform density on the simplex. For iteration i, draw π^{U} from a Dirichlet with parameter $\alpha \propto 1_{C\times 1}$ for the first iteration and $\alpha \propto (\pi^{\mathrm{U}})_{i-1}$ for subsequent iterations.
Chronological π^{L}, Uniform Random π^{U}	Draw the labeled set chronologically. Then, draw π^{U} by selecting a random point on the simplex.
Extreme Proportion Shift	Select division that best approximates one of the categories having $< 5\%$ of the labeled set, but $> 25\%$ of the unlabeled set.
Uniform Random Proportions	Draw π^{L} and π^{U} from independent uniform distributions on the simplex.
Extreme Features in Labeled Set	Calculate document-level word vector features. Form the labeled set from documents falling furthest from the average document. Form the unlabeled set from a random selection of the remaining documents.

Appendix C Applications to Causal Inference

Some of the innovations developed here might be profitably applied to areas outside of automated text analysis. For example, our dimensionality reduction technique may have applications to matching for causal inference, from which we borrowed inspiration for some of our ideas. An important issue in matching is the optimal feature space, such as in the space of predicted values for the outcome under the control intervention (such as in "predictive mean matching"). One such feature space is the one derived here which might enable researchers to control dimensionality, as well as the tradeoff between feature independence and informativeness. The resulting causal estimator could thus have attractive properties, since it would take into account both the relationship between the covariates and the outcome (leading to lower bias) while incorporating several independent sources of information (leading to lower variance).

To be more precise, consider the Average Treatment Effect on the Treated:

$$\tau = E[Y_i(1)|X_i, T_i = 1] - E[Y_i(0)|X_i, T_i = 1].$$

We could estimate $E[Y_i(0)|X_i, T_i = 1]$ directly using a regression model trained on the control units, predicting their outcomes. However, to avoid model dependence we could instead use non-parametric methods, matching each treated unit with the nearest (say) 3 control units and taking the average outcome value of those control units as an estimate of $E[Y_i(0)|X_i, T_i = 1]$. However, some variables on which we match may be unimportant, in that they are poorly predictive of the outcome. When outcomes are discrete, we can consider applying the technique developed above to generate feature projections for readme to instead create features for matching that are highly predictive of the outcome. We could do this by fitting a feature projection on units in the control group (to avoid violating the Tied Hands Principle) that optimizes the degree of feature discrimination in Y_i. We can then apply this projection to all units and match on the new feature space.

In this arrangement, the background covariates function like the word vector summaries in our text analysis, and $Y_i(0)$ plays a role that category membership did before. Matching on features that generate a high-quality $E[\underline{\mathbf{X}}_i|Y_i(0)]$ matrix may seem strange. Yet, the exercise here is in many ways the natural one: either we make no assumptions about the covariate-outcome relationship (as in fully non-parametric matching), we condition on the background covariates and use this information to predict the outcome (as in the case of regression-adjusted inference), or we condition on the outcome data and re-weight the background covariates in a way that maximizes the distinctiveness of the resulting features (as in our proposed approach).

As a proof of concept, we conduct simulations beginning with only the control units from each of 9 prominent social science experiments (Gerber and Green, 2000; McClendon and Riedl, 2015; Enos and Fowler, 2016; Taylor, Stein, Woods, and Mumford, 2011; Bailey, Hopkins, and Rogers, 2016; Finkelstein et al., 2012; Kugler, Kugler, Saavedra, and Prada, 2015; Bolsen, Ferraro, and Miranda, 2013; Calle, De Mel, McIntosh, and Woodruff, 2014). In each replication data set, we assign half of the control units to receive a "treatment" of a size equal to a constant plus 0.1 times the standard deviation of the outcome (which we take to be the original dependent variable in the study). We assign this synthetic treatment probabilistically in a way that seeks to replicate the amount of confounding present in the original experiment. In particular, after fitting a non-parametric

binary classifier to the original treatment/control status, we selected units into the synthetic treatment group with probabilities proportional to the resulting propensity scores.

We then compare the error of 5 different methods for extracting the causal effect between the synthetic treatment and control groups. These methods are each based on nearest neighbor matching and differ only in the feature space in which the matching takes place. We consider matching on estimated propensity scores, on random projections, and on the predicted values from a regression modeling the relationship between the data inputs and the outcome trained on the control data. We also considered matching on the original features, on random projections, and on the features generated from the readme2 algorithm. We repeated the synthetic experiment 100 times for each of the 9 replication data sets.

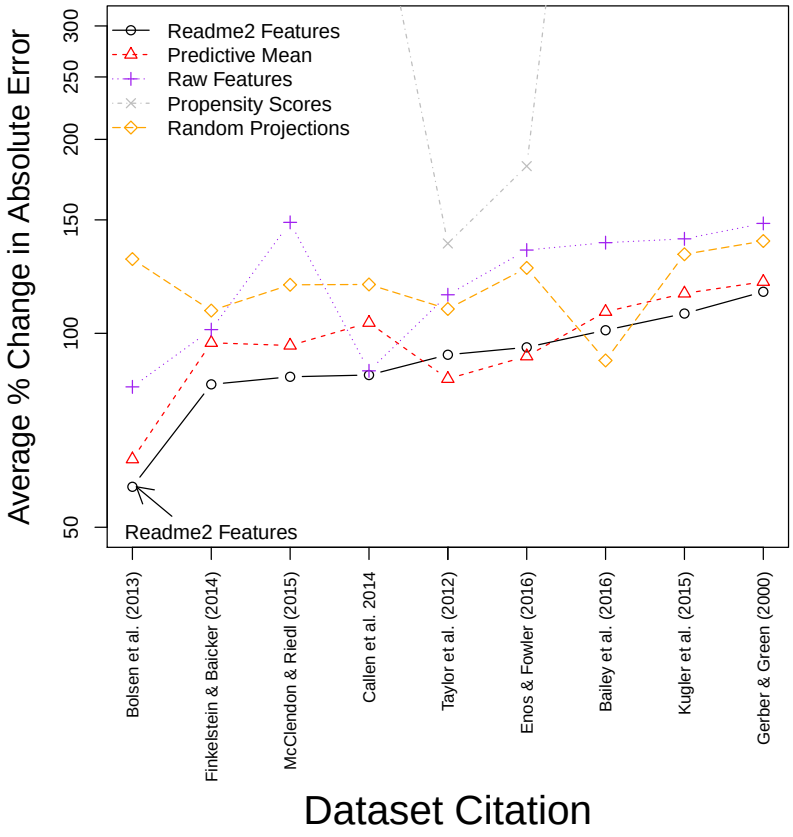

Figure 8: The change in absolute error in this graph is calculated relative to the error from the simple difference in means estimator. Points below "100" are those for which the average error of the matching estimator was lower than the naive difference in means estimator. Points above "100" are those for which the error was higher.

The results, which appear in Figure 8, suggest that the readme2 features perform well for the non-parametric estimation of causal effects. In 7 of the 9 data sets, estimation in the readme2 space yields the largest decrease in absolute error. In the other 2 cases, performance is still very good. It appears that the features from the readme2 algorithm can be profitably used in other contexts where non-parametric analyses will be done and

where outcome data is available.[5]

References

Abadi, Martin et al. (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. URL: tensorflow.org.

Bailey, Michael, Daniel Hopkins, and Todd Rogers (2016). "Unresponsive and Unpersuaded: The Unintended Consequences of a Voter Persuasion Effort". In: *Political Behavior* 3, pp. 713–746.

Bolsen, Toby, Paul Ferraro, and Juan Jose Miranda (Aug. 2013). "Are Voters More Likely to Contribute to Other Public Goods? Evidence from a Large-Scale Randomized Policy Experiment". In: *ajps* 58.1, pp. 17–30.

Brunzell, H. and J. Eriksson (2000). "Feature reduction for classification of multidimensional data". In: *Pattern Recognition* 33.10, pp. 1741–1748. DOI: 10.1016/S0031-3203(99)00142-9. URL: bit.ly/2ihoYdl.

Buck, Alfred A and John J Gart (1966). "Comparison of a Screening Test and a Reference Test in Epidemiologic Studies. I. Indices of Agreements and their Relation to Prevalence." In: *American Journal of Epidemiology* 83.3, pp. 586–92.

Calle, Michael, Suresh De Mel, Craig McIntosh, and Christopher Woodruff (2014). *What are the Headwaters of Formal Savings? Experimental Evidence from Sri Lanka*. NBER.

Ceron, Andrea, Luigi Curini, and Stefano M. Iacus (2016). "iSA: A fast, scalable and accurate algorithm for sentiment analysis of social media content". In: *Information Sciences* 367, pp. 105–124.

Denny, Matthew James and Arthur Spirling (2016). "Assessing the Consequences of Text Preprocessing Decisions". In: https://ssrn.com/abstract=2849145.

Enos, Ryan and Anthony Fowler (Apr. 2016). "Aggregate Effects of Large-Scale Campaigns on Voter Turnout". In: *Political Science Research and Methods* 21, pp. 1–19.

Esuli, Andrea and Fabrizio Sebastiani (2015). "Optimizing text quantifiers for multivariate loss functions". In: *ACM Transactions on Knowledge Discovery from Data (TKDD)* 9.4, p. 27.

Finkelstein, Amy et al. (Aug. 2012). "The Oregon Health Insurance Experiment: Evidence from the First Year". In: *The Quarterly Journal of Economics* 127.3, pp. 1057–1106.

Firat, Aykut (2016). "Unified Framework for Quantification". In: *arXiv:1606.00868*.

Forman, George (2007). *Quantifying counts, costs, and trends accurately via machine learning*. Tech. rep. Technical report, HP Laboratories, Palo Alto, CA.

Gerber, Alan S. and Donald P. Green (Sept. 2000). "The Effects of Canvassing, Telephone Calls, and Direct Mail on Voter Turnout: A Field Experiment". In: *American Political Science Review* 94.3, pp. 653–663.

Hand, David J. (2006). "Classifier Technology and the Illusion of Progress". In: *Statistical Science* 21.1, pp. 1–14.

[5] Our dimensionality reduction technology may also be used for data visualization. For example, in visualizing data on partisanship, we could find the 2-dimensional projection that maximally discriminates between Democrats, Republicans, and Independents and simultaneously contains minimal redundancy. The relevant clusters would then become more visible, and could even be paired with a data clustering algorithm on the 2-dimensional projection for additional visualization or analysis purposes.

Ho, Daniel E., Kosuke Imai, Gary King, and Elizabeth A. Stuart (2007). "Matching as Nonparametric Preprocessing for Reducing Model Dependence in Parametric Causal Inference". In: *Political Analysis* 15, pp. 199–236. URL: j.mp/matchP.

Hoadley, Bruce (2001). "[Statistical Modeling: The Two Cultures]: Comment". In: *Statistical Science* 16.3, pp. 220–224.

Hopkins, Daniel and Gary King (Jan. 2010). "A Method of Automated Nonparametric Content Analysis for Social Science". In: *American Journal of Political Science* 54.1, pp. 229–247. URL: j.mp/jNFDgI.

Hopkins, Daniel, Gary King, Matthew Knowles, and Steven Melendez (2013). *Readme: Software for Automated Content Analysis*. Versions 2007–2013. URL: GaryKing.org/readme.

Iacus, Stefano M., Gary King, and Giuseppe Porro (2012). "Causal Inference Without Balance Checking: Coarsened Exact Matching". In: *Political Analysis* 20.1, pp. 1–24. URL: j.mp/woCheck.

James, Gareth, Daniela Witten, Trevor Hastie, and Robert Tibshirani (2013). *An introduction to statistical learning*. Vol. 112. Springer.

Kar, Purushottam et al. (2016). "Online Optimization Methods for the Quantification Problem". In: *arXiv preprint arXiv:1605.04135*.

King, Gary, Daniel Hopkins, and Ying Lu (2012). "System for estimating a distribution of message content categories in source data". US Patent 8,180,717. URL: j.mp/VApatent.

King, Gary and Ying Lu (2008). "Verbal Autopsy Methods with Multiple Causes of Death". In: *Statistical Science* 23.1, pp. 78–91. URL: j.mp/2AuA8aN.

King, Gary, Ying Lu, and Kenji Shibuya (2010). "Designing Verbal Autopsy Studies". In: *Population Health Metrics* 8.19. URL: j.mp/DAutopsy.

King, Gary, Jennifer Pan, and Margaret E. Roberts (2013). "How Censorship in China Allows Government Criticism but Silences Collective Expression". In: *American Political Science Review* 107, pp. 1–18. URL: j.mp/LdVXqN.

Kingma, Diederik and Jimmy Lei Ba (2015). "Adam: A Method for Stochastic Optimization". In: *Proceedings of the International Conference on Learning Representations*. International Union for the Scientific Study of Population.

Kugler, Adriana, Maurice Kugler, Juan Saavedra, and Luis Omar Herrera Prada (2015). *Long-term Direct and Spillover Effects of Job Training: Experimental Evidence from Colombia*. NBER.

Levy, Omer, Yoav Goldberg, and Ido Dagan (2015). "Improving distributional similarity with lessons learned from word embeddings". In: *Transactions of the Association for Computational Linguistics* 3, pp. 211–225.

Levy, P.S. and E. H. Kass (1970). "A three population model for sequential screening for Bacteriuria". In: *American Journal of Epidemiology* 91, pp. 148–154.

McClendon, Gwyneth and Rachel Beatty Riedl (Oct. 2015). "Religion as a stimulant of political participation: Experimental evidence from Nairobi, Kenya". In: *jop* 77.4, pp. 1045–1057.

Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean (2013). "Efficient estimation of word representations in vector space". In: *arXiv preprint arXiv:1301.3781*.

Mikolov, Tomas, Wen-tau Yih, and Geoffrey Zweig (2013). "Linguistic regularities in continuous space word representations". In: *Proceedings of the 2013 Conference of*

the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 746–751.

Milli, Letizia et al. (2013). "Quantification trees". In: *2013 IEEE 13th International Conference on Data Mining*, pp. 528–536.

Pennington, Jeffrey, Richard Socher, and Christopher Manning (2014). "Glove: Global vectors for word representation". In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543.

Pereyra, Gabriel et al. (2017). "Regularizing Neural Networks by Penalizing Confident Output Distributions". In: URL: https://arxiv.org/abs/1701.06548.

Rumelhart, David E, Geoffrey E Hinton, and Ronald J Williams (1986). "Learning representations by back-propagating errors". In: *Nature* 323.6088, p. 533.

Socher, Richard et al. (2013). "Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 1631–1642.

Srivastava, Nitish et al. (2014). "Dropout: a simple way to prevent neural networks from overfitting". In: *The Journal of Machine Learning Research* 15.1, pp. 1929–1958.

Tasche, Dirk (2016). "Does quantification without adjustments work?" In: *arXiv preprint arXiv:1602.08780*.

Taylor, Bruce, Nan Stein, Dan Woods, and Elizabeth Mumford (Oct. 2011). *Shifting Boundaries: Final Report on an Experimental Evaluation of a Youth Dating Violence Prevention Program in New York City Middle Schools*. Final Report. Police Executive Research Forum.

Vincent, Pascal et al. (2010). "Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion". In: *Journal of Machine Learning Research* 11.Dec, pp. 3371–3408. URL: bit.ly/2gPcedw.

Zhang, Chiyuan et al. (2016). "Understanding deep learning requires rethinking generalization". In: *arXiv preprint arXiv:1611.03530*.