Featurizing Text: Converting Text into Predictors for Regression Analysis

Dean P. Foster* Mark Liberman Robert A. Stine*

Department of Statistics

The Wharton School of the University of Pennsylvania

Philadelphia, PA 19104-6340

October 18, 2013

Abstract

Modern data streams routinely combine text with the familiar numerical data used in regression analysis. For example, listings for real estate that show the price of a property typically include a verbal description. Some descriptions include numerical data, such as the number of rooms or the size of the home. Many others, however, only verbally describe the property, often using an idiosyncratic vernacular. For modeling such data, we describe several methods that that convert such text into numerical features suitable for regression analysis. The proposed featurizing techniques create regressors directly from text, requiring minimal user input. The techniques range naive to subtle. One can simply use raw counts of words, obtain principal components from these counts, or build regressors from counts of adjacent words. Our example that models real estate prices illustrates the surprising success of these methods. To partially explain this success, we offer a motivating probabilistic model. Because the derived regressors are difficult to interpret, we further show how the presence of partial quantitative features extracted from text can elucidate the structure of a model.

Key Phrases: sentiment analysis, n-gram, latent semantic analysis, text mining

*Research supported by NSF grant 1106743

1 Introduction

Modern data streams routinely combine text with numerical data suitable for in regression analysis. For example, patient medical records combine lab measurements with physician comments and online product ratings such as those at Amazon or Netflix blend explicit characteristics with verbal commentary. As a specific example, we build a regression model to predict the price of real estate from its listing. The listings we use are verbal rather than numerical data obtained by filling out a spreadsheet-like form. Here are four such listings for Chicago, IL, extracted (with permission) from trulia.com on June 12, 2013:

```
$399000 Stunning skyline views like something from a postcard are yours
with this large 2 bed, 2 bath loft in Dearborn Tower!  Detailed
hrdwd floors throughout the unit compliment an open kitchen and
spacious living-room and dining-room /w walk-in closet, steam
shower and marble entry.  Parking available.

$13000 4 bedroom, 2 bath 2 story frame home. Property features a
large kitchen, living-room and a full basement. This is a Fannie Mae
Homepath property.

$65000 Great short sale opportunity...  Brick 2 flat with 3 bdrm
each unit. 4 or more cars parking. Easy to show.

$29900 This 3 flat with all 3 bed units is truly a great
investment!! This property also comes with a full attic that has
the potential of a build-out-thats a possible 4 unit building in a
great area!!  Blocks from lake and transportation. Looking for a
deal in todays market - here is the one!!!
```

The only numerical data common to the listings is the price that appears at the head of each listing. Some listings include further numerical data, such as the number of rooms or occasionally the size of the property (number of square feet). Many listings, however, provide only a verbal description, often written in an idiosyncratic vernacular familiar only to those who are house hunting. Some authors write in sentences, others not, and a variety of abbreviations appear. The style of punctuation varies from spartan to effusive (particularly exclamation marks), and the length of the listing runs from several words to a long paragraph.

An obvious approach to building regressors from text data relies on a substantive analysis of the text. For example, sentiment analysis constructs a domain-specific lexicon of positive and negative words. In the context of real estate, words such as ‘modern’ and ‘spacious’ might be flagged as positive indicators (and so be associated with more expensive properties), whereas ‘Fannie Mae’ and ‘fixer-upper’ would be marked as negative indicators. The development of such lexicons has been an active area of research in sentiment analysis over the past decade (Taboada, Brooke, Tofiloski, Voli and Stede, 2011). The development of a lexicon require substantial knowledge of the context and the results are known to be domain specific. Each new problem requires a new lexicon. The lexicon for pricing homes would be quite different from the lexicon for diagnosing patient health. Our approach is also domain specific, but requires little user input and so can be highly automated.

In contrast to substantively oriented modeling, we propose a version of supervised sentiment analysis that converts text into conventional explanatory variables. We convert the text into conventional numerical regressors (featurize) by exploiting methods from computational linguistics that are familiar to statisticians. These so-called vector space models (Turney and Pantel, 2010), such as latent semantic analysis (LSA), make use of singular value decompositions of the bag-of-words and bigram representations of text. (This connection leads to methods being described as a ‘spectral algorithm for’.) These representations map words into points in a vector space defined by counts. This approach is highly automated with little need for human intervention, though it makes it easy to exploit such investments when available. The derived regressors can be used alone or in combination with traditional variables, such as those obtained from a lexicon or other semantic model. We use the example of real estate listings to illustrate the impact of various choices on the predictive accuracy. For example, a regression using the automated features produced by this analysis explains over two-thirds of the variation in listed prices for real estate in Chicago. The addition of several substantively derived variables adds little. Though we do not emphasize its use here, variable selection can be employed to reduce the ensemble of regressors without sacrificing predictive accuracy.

Our emphasis on predictive accuracy does not necessarily produce an interpretable model, and one can use other data to create such structure. Our explanatory variable resemble those from principal components analysis and share their anonymity. To provide more interpretable regressors, the presence of partial quantitative information in real estate listings (*e.g.*, some listings include the number of square feet) provides what we call lighthouse variables that can be used to derive more interpretable variables. In our sample, few listings (about 6%) indicate the number of square feet. With so much missing data, this manually derived predictor is not very useful as an explanatory variable in a regression. This partially observed variable can then be used to define a weighted sum of the anonymous text-derived features, producing a regressor that is both complete (no missing cases) and interpretable. One could similarly use features from a lexicon to provide more interpretable features.

The remainder of this paper develops as follows. The following section provides a concise summary of our technique. The method is remarkably simple to describe. Section 3 demonstrates the technique using about 7,500 real estate listings from Chicago. Though simple to describe, it is more subtle to appreciate why it works. Our explanation appears in Section 4 which shows how this technique discovers the latent effects in a topic model for text. We return to models for real estate in Section 5 with a discussion of the use of variable selection methods ands use cross-validation to measure the success of methods and to compare several models. Variable selection is particularly relevant if one chooses to search for nonlinear behavior. Section 6 considers the use of partial semantic information for producing more interpretable models. We close in Section 7 with a discussion and collection of future projects. Our aim is to show how easily one can convert text into familiar regressors for regression. As such, we leave to others the task of attempting to explain why such simple representations as the co-occurrence of words in documents might capture the deeper meaning (Deerwester, Dumais, Furnas, Landauer and Harshman, 1990; Landauer and Dumais, 1997; Bullinaria and Levy, 2007; Turney and Pantel, 2010).

2 An Algorithm for Featurizing Text

Our technique for featurizing text has 3 main steps. These steps are remarkably simple:

1. Convert the source text into lists of word types. A word type is a unique sequence of non-blank characters. Word types are not distinguished by meaning or use. That is, this analysis does not distinguish homographs.

2. Compute matrices that (a) count the number of times that word types appear within each document (such as a real estate listing) and (b) count the number of times that word types are found adjacent to each other.

3. Compute truncated singular value decompositions (SVD) of the resulting matrices of counts. The leading singular vectors of these decompositions are our regressors.

The simplicity of this approach means that this algorithm runs quickly. The following analysis of 7,384 real-estate listings generates 1,000 features from raw text in a few seconds on a laptop. The following paragraphs define our notation and detail what happens within each step.

The process of converting the source text into word tokens, known as tokenization, is an easily overlooked, but critical step in the analysis. A word token is an instance of a word type, which is roughly a unique sequence of characters delimited by white space. We adopt a fairly standard, simple approach to converting text into tokens. We convert all text to lower case, separate punctuation, and replace rare words by an invariant "unknown" token. To illustrate some of the issues in converting text into tokens, the following string is a portion of the description of a property in Chicago:

```
Brick flat, 2 bdrm.  With two-car garage.
```

Separated into tokens, this text becomes a list of 10 tokens representing 9 word types:

```
{brick, flat, <,>, 2, bdrm, <.>, with, two-car, garage,<.>}
```

Once tokenized, all characters are lower case. Punctuation symbols, such as commas and periods, are "words" in this sense. We leave embedded hyphens in place. Since little is known about rare words that are observed in only one or two documents, we represent their occurrence by the symbol '<UNK>'. The end of each document is marked by a unique type. We make no attempt to correct spelling errors and typos nor to expand abbreviations. References such as the books Manning and Schütz (1999) and Jurafsky and Martin (2009) describe further processing, such as stemming and annotation that can be done prior to statistical modeling. Turney and Pantel (2010) gives a concise overview.

Once the source text has been tokenized, we form two matrices of counts. The SVD of each of these defines a set of explanatory variables. The matrices, W and B, differ in how they measure the similarity of words. Words are judged to be similar if they appear in the same context. For the document/word matrix W, the context is a document – a real estate listing. This matrix holds counts of which words appear in the same document, ignoring the order in which the words appear. This approach treats each document (or listing) as a bag of words, a multiset that does not distinguish the placement of the words. The second matrix adopts a very different perspective that relies entirely upon ordering; it defines the context by adjacency. The bigram matrix B counts how often words appear adjacent to each other. The document/word and bigram matrices thus represent two extremes of a common approach: Associate words that co-occur within some context. W uses the wide window provided by a document, whereas B uses the most narrow window possible. The wider window afforded by a document hints that W emphasizes semantic similarity, whereas the narrow window of adjacency that defines B suggests more emphasis on local syntax. Curiously, we find either approach effective and make use of both.

Associating words that co-occur in a document is more familiar to statisticians, and so we begin there. Let V denote a vocabulary consisting of M unique word types. The vector w_i holds the counts of these word types for the ith document; w_{im} is the number of times word type m appears within the ith document. (All vectors in our notation are column vectors.) Let n denote the number of documents; these documents are the observational units in our analysis. Fro models of real estate, a document is the description found in a listing. We collect the word counts for documents as rows within the $n \times M$ matrix W,

$$W = \begin{pmatrix} w_1' \\ w_2' \\ \vdots \\ w_n' \end{pmatrix}$$

(Note that within computational linguistics it is common to find the transpose of this matrix.) The matrix W is quite sparse: most documents use a small portion of the vocabulary. Let $m_i = \sum_m w_{im}$ denote the number of word tokens that appear in the description of the ith property. It is common within linguistics to transform these counts prior to additional modeling. For example, the counts might be normalized by document, or transformed to emphasize relatively rare events. Turney and Pantel (2010) summarizes several approaches, such as the popular TF-IDF (term frequency-inverse document frequency) and entropy-based transformations.

The bigram matrix counts how often word types occur adjacent to each other. Let B define the $M \times M$ matrix produced from the sequence of tokens for all documents combined (the corpus). B_{ij} counts how often word-type i precedes work-type j within the corpus. Whereas W ignores word placement (sequencing) within a document, B combines counts over all documents and relies on the sequence of word tokens.

We obtain regression features from the SVD of W and B. The regressors are immediate from the SVD of W. Let

$$W = \tilde{U}_W \tilde{D}_W \tilde{V}_W' \tag{1}$$

denote the SVD of W. We typically use only a subset of this decomposition, and so we define U_W to be the $n \times k_W$ matrix defined by the leading k_W singular vectors of W (*i.e.*, the first k_W columns of $\tilde{U}_W$ that are associated with the largest singular values). The collection U_W defines a collection of regressors. (The resulting computation that isolates only these leading singular vectors is sometimes called a truncated SVD.) This representation of text is known as latent semantic analysis (or latent semantic indexing) within computational linguistics; statisticians will recognize this as a principal components analysis (PCA) of W. The choice of the number of singular vectors to retain, k_W, is a user-controlled tuning parameter of this technique. We will provide some advice on unsupervised methods for picking k_W in the following section within the example that analyzes real estate listings in Chicago.

A second application of the SVD produces regressors from the bigram matrix B. Let

$$B = \tilde{U}_B \tilde{D}_B \tilde{V}_B' \tag{2}$$

define the SVD of B, and again use matrices without $\sim$ to denote components of the truncated SVD: U_B and V_B denote the first k_B columns of $\tilde{U}_B$ and $\tilde{V}_B$, respectively. As in the decomposition of W, the number of singular vector to retain is a user-defined choice. We generally keep $k_W = k_B$. Because B is $M \times M$, these singular vectors define points in $\mathbb{R}^M$ and are sometimes referred to as "eigenwords" because of the way in which they form directions in word space **{ras:** *cite*}. The ith row of U_B locates the word type w_i within $\mathbb{R}^{k_B}$ (all of the following applies to V_B as well). To build regressors, we locate each document at a point within this same space. We can think of this location in two, nearly equivalent ways that emphasize either the rows or columns of U_B. The two methods differ in a sum is normalized. The row-oriented motivation is particularly simple: a document is positioned at the average of the positions of its words. For example, the ith document is located at $w_i' U_B / m_i$. Alternatively, emphasizing columns, we can compute the correlation between the columns of U_B with the bag-of-words representations of the documents. Because the columns of U_B and V_B are orthonormal, these correlations are given by

$$C = [C_l C_r] = \mathrm{diag}(\|w_i\|^{-1}) W [U_B \; V_B] \,, \quad \text{where} \quad \|x\|^2 = \sum x_i^2 \,. \tag{3}$$

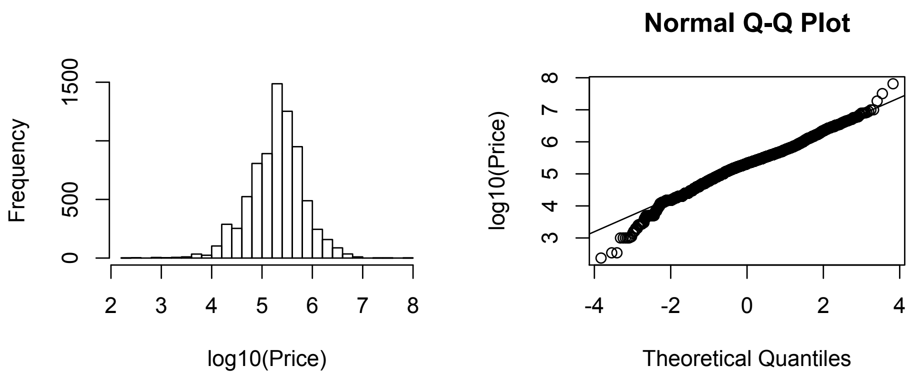
The ith row of of the $n \times 2k_B$ matrix C is the vector of correlations between the bag-of-words representation w_i and the singular vectors of B. In our models for real-estate listings, the columns of C form the second bundle of regressors.

It is worthwhile to take note of two properties of these calculations that are important in practice. First, one needs to take advantage of sparsity in the matrices W and B to reduce memory usage and to increase the speed of computing matrix products. Second, the computation of the SVD of a large matrix can be quite slow. For example, computing the SVD of B is of order $O(M^3)$ and one can easily have vocabulary of M =10,000 or more word types. To speed this calculation, we exploit random projection algorithms defined and analyzed in Halko, Martinsson and Tropp (2010).

3 Predicting Prices of Real Estate

This section demonstrates the use of regressors defined from text using the featurizing techniques defined in the prior section. The data are n =7,384 property listings for Chicago, IL in June, 2013. (Note that at the time, trulia.com showed 30,322 listings for

Figure 1: *The distribution of prices for real estate in Chicago is highly skewed, but a log transformation produces data that are nearly normal.* The presence of a long lower tail might indicate that the data mix typical home sales with special, subsided properties or perhaps vacant lots that sell for an unusually low price.

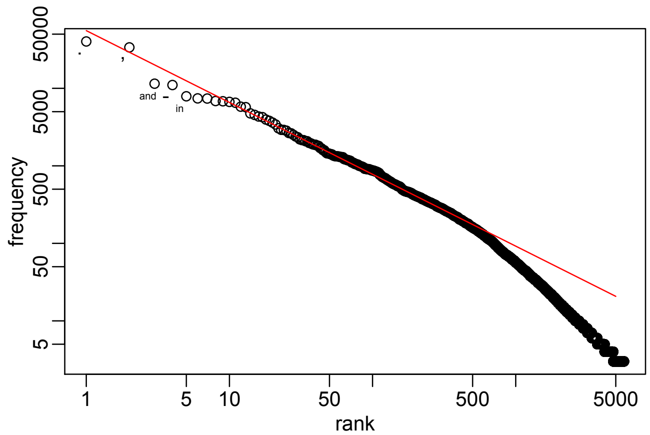


Chicago, but most of these were foreclosures that are excluded in our analysis.) The response in our models is the log of the listed price. The prices for properties listed in Chicago is quite skewed, so we transformed the response to a log scale as shown in the histogram of Figure 1. This display uses the base 10 log of the prices for easy interpretation; subsequent models use natural logs throughout. The log transformation produces a roughly Gaussian distribution, with a hint of fat tails most prominently for cheaper properties with prices near $25,000.

3.1 Tokenization and Parsing

The 7,384 property listings in Chicago contain 543,869 word tokens that define 15,228 word types. More than half of these tokens appear only once or twice, providing little exposure to how the word is used. We clustered these rare types into one category (<UNK>), resulting in a reduced vocabulary of $M = 5,708$ word types. The most common word types are "not words" but rather punctuation: '.' occurs 40,227 times and ',' 33,746 times. Following these come the collection of seldom seen words (OOV, 11,478), 'and' (frequency 11,032), '!' (7,894) and 'in' (7,418). As usual in text, "most types are common but most words are rare." That is, the most common types occur

Figure 2: *The log of the frequency of word types in the listings for properties in Chicago is roughly linear in the log of the rank of the words, a Zipf distribution.* The shown line $\log \text{freq} = 10.9 - 0.93 \log \text{rank}$ was fit to words with ranks 1 through 500.

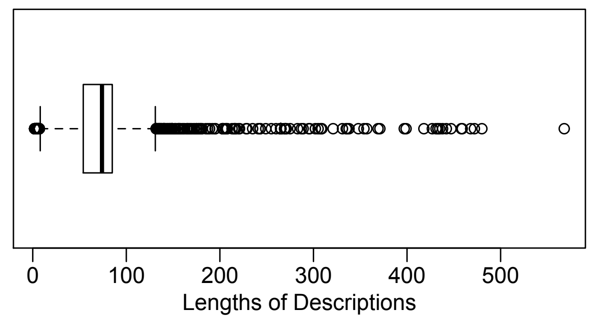


frequently whereas most words appear infrequently. Figure 2 shows a scatterplot of the log of the frequency of these word types versus the log of their ranks. It is common in text to find a linear trend with slope near 1 in this graph, a Zipf distribution (Zipf, 1935; Baayen, 2002). Even though this text is not standard English, one expects to find counts resembling those produced by a power law (Clauset, Shalizi and Newman, 2009). For this text, the shown line (with log-log slope -0.927) holds only for more common word types. For less common words (here, outside the most common 500 words), the frequencies drop off more rapidly. (This change in slope is also seen for words in Wikipedia.)

The average description has about 73 word tokens, but the distribution of the lengths m_i is right skewed. The boxplot in Figure 3 shows the lengths. The shortest description has 2 tokens, whereas the longest description has 568 tokens. This variation in the lengths of the descriptions suggests that modeling the prices from this text will require a weighted regression. We simply do not know so much about properties with short descriptions.

Before we use the decomposition rules to construct regressions, we construct several by extracting explicit values from the descriptions. For example when processing ads for real estate, one might conjecture that an agent has a lot more to say in when describing

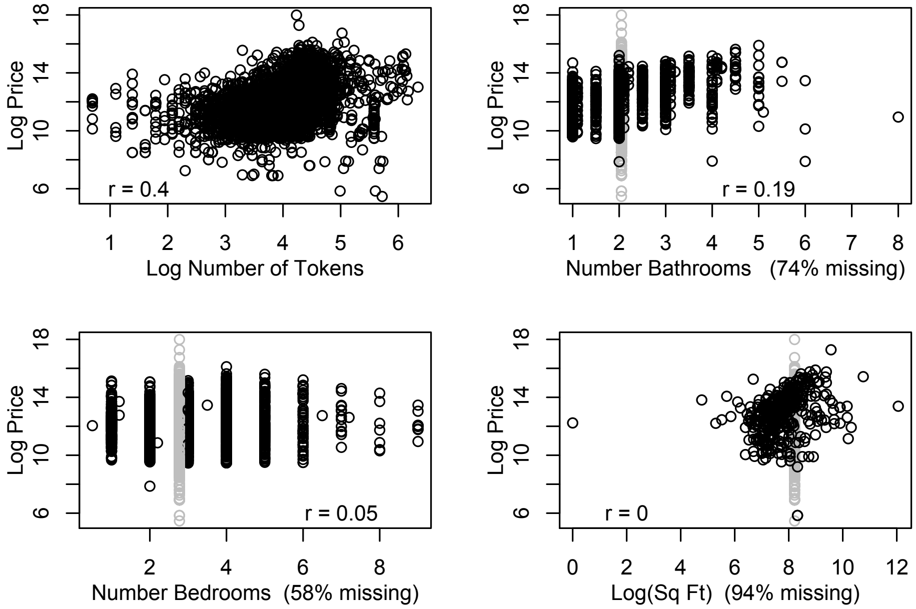
Figure 3: *The distribution of the lengths m_i of the property descriptions is right-skewed, with some listing running hundreds of words compared to a median length of 74 words.*



the features of an expensive property than a property in need of repair, implying that the length of the ad for a property would m_i may be predictive of the price. In addition, one can use a regular expression to extract the number of bedrooms when it appears in the listing. We wrote regular expressions to extract numerical data from advertisements when present. Constructing these is a labor-intensive process that must be done on a case-by-case basis. For example, a regular expression would parse the value 2 for the number of bedrooms and bathrooms from the first listing shown in the introduction. The patterns used in these regular expressions allow common abbreviations, such as "bth", "bath" and "bthrm" for bathrooms. Another regular expression that we used extracts the number of square feet from the listing. Most listings, however, omit these characteristics. For Chicago, our regular expressions found that 6% of the listings indicate the number of square feet, 26% indicate the number of bathrooms, and 42% give the number of bedrooms. More complex regular expressions would likely find more matches, but the gains are likely to be few. One also faces a Type I/Type II trade-off. Simple regular expressions omit some matches, but more aggressive expressions match inadvertently. To accommodate the large amount of missing data, we used the simple procedure of adding indicator variables that distinguish observed cases for these variables, and we then filled the missing values with means.

The four scatterplots in Figure 4 summarize the marginal association between the log of prices and these parsed predictors, including the number of words in descriptions. Missing values produce the columns of gray points located at the mean of the variable

Figure 4: *The parsed characteristics have slight positive association with the log of the prices.* Gray points in the figures identify cases that were missing the explanatory variable; the shown correlation uses the complete, filled-in data.



on the x axis. None of these features is highly correlated with the log of price; the highest correlations are with the length of the description ($r = 0.40$) and the number of bathrooms ($r = 0.19$). One explanation for the low association is the abundance of missing data. For example, corr(log price, log sq ft) = 0.00 overall, but is larger (0.26) among the few properties that report this characteristic. These plots also show several anomalies. For example, the first scatterplot of the log of price on the number of words shows a cluster of 25 listings, all with exactly 265 words. All of these different properties were listed in a common format by a government agency. The scatterplot of the log of prices on the log of the square footage also shows a clear outlier; this outlier is a consequence of aggressive parsing. A typo in a description ("has 1sfam") led to the regular expression finding a property with 1 square foot.

Table 1 summarizes the fit of the regression of the log of price on these four extracted variables and the missing indicators. In this first example, the regressors are not highly correlated, and the importance of these variables in the multiple regression generally echoes the marginal correlations shown in Figure 4. Among the seven explanatory

Table 1: *OLS multiple regression of log prices on the parsed explanatory variables and indicators of observed values.*

	Estimate	Std. Error	t value	Pr(>|t|)
Intercept	4.8858	0.4944	9.88	0.0000
$\log m$	0.8211	0.0239	34.31	0.0000
log Sq Ft	0.3680	0.0592	6.21	0.0000
Sq Ft Obs	0.6108	0.0607	10.06	0.0000
Bedrooms	-0.0095	0.0169	-0.56	0.5734
Bedroom Obs	0.0023	0.0306	0.08	0.9400
Bathrooms	0.4153	0.0307	13.53	0.0000
Bathroom Obs	0.0038	0.0344	0.11	0.9127

$s_e = 1.084$ with $R^2 = 0.193$, $\overline{R}^2 = 0.192$

variables, the length of the descriptions is most predictive, followed by the number of bathrooms. Unlike the marginal associations, however, both the log of the number of square feet (when observed) and its missing indicator are significant. The missing indicators for bedrooms and bathrooms are not predictive. The model obtains adjusted R-squared $\overline{R}^2 = 0.192$, with residual standard deviation $s_e = 1.08$. Because we compare regression models with varying numbers of explanatory variables, $\overline{R}^2$ is more useful than without adjusting for degrees of freedom. Since we are not doing variable selection at this point, but rather fitting full ensembles of pre-selected variables, adjusted R-squared is an unbiased summary of each model's performance out of sample. (Section 5 reports cross-validation results for these models.)

Before building models using singular vectors, we begin by regressing the response on indicators for the words. That is, we simply regress y on the word counts in W. This model provides a baseline for comparison with the fits produced by regressions using singular vectors. Table 2 summarizes the fit using the most common 2,000 words. Overall, this model produces $\overline{R}^2 = 0.681$. Residual plots show fat tails (consistent with that in the prices, Figure 1) but no clear evidence of heteroscedasticity that might be anticipated due to the differences in the document lengths (Figure 3). The most significant word is "vacant" with $t = -8.5$; not surprisinging, the presence of this word in a listing indicates a property with lower than usual value. In contrast, out-of-vocabulary words denoted "OOV" have higher than average prices ($t = 6.3$). Figure 5 summarizes the distribution of all t-statistics in this model. Twenty-two words are not used due to singularities among these counts, leaving 1,978 estimated coefficients. The dashed line in the scatterplot of $|t_j|$ on j in the left panel of Figure 5 is the Bonferroni threshold $\Phi^{-1}(1 - 0.025/1978) \approx 4.21$. Only 14 estimates exceed this threshold for statistical significance. The nearly flat red line in the figure is a lowess smooth of the $|t_j|$. The half-normal plot in the right panel of Figure 5 confirms the diffuse signal in this model: the distribution of the fitted t-statistics is not far from the null distribution. The gray line in the half-normal plot is the diagonal; the red line is the fitted regression of the smallest 200 $|t|$-statistics on the corresponding quantiles. For a sample from a standard normal distribution, the slope of a line fit to the t values should be 1. The slope of the fitted line for these estimates is significantly larger, but clearly the signal is widely spread over these estimates. A regression on counts for 14 words whose estimates exceed the Bonferroni bound (Table 2) obtains $\overline{R}^2 = 0.191$. Thus, this estimation problem differs from the so-called "nearly black" models often studied in research on variable selection. In those models, a few estimates stand out from a noisy background. In this application, much of the signal lies embedded in that background.

The next model uses regressors created from the SVD of the document/word count matrix W defined in equation (1). We retained $k_W = 100, 200, \ldots, 500$ singular vectors of W. Our analysis suggests each of these collections of singular vectors both retains too many insignificant features while at the same time omits others that are predictive. Broadly speaking, the leading singular vectors (those with larger singular values) are more predictive than subsequent singular vectors. That said, not all of the leading vectors are statistically significant nor do all of the later singular vectors have coefficients near zero. As a result, variable selection from a yet larger collection of singular vectors may provide a better fit, a task we defer to Section 5.

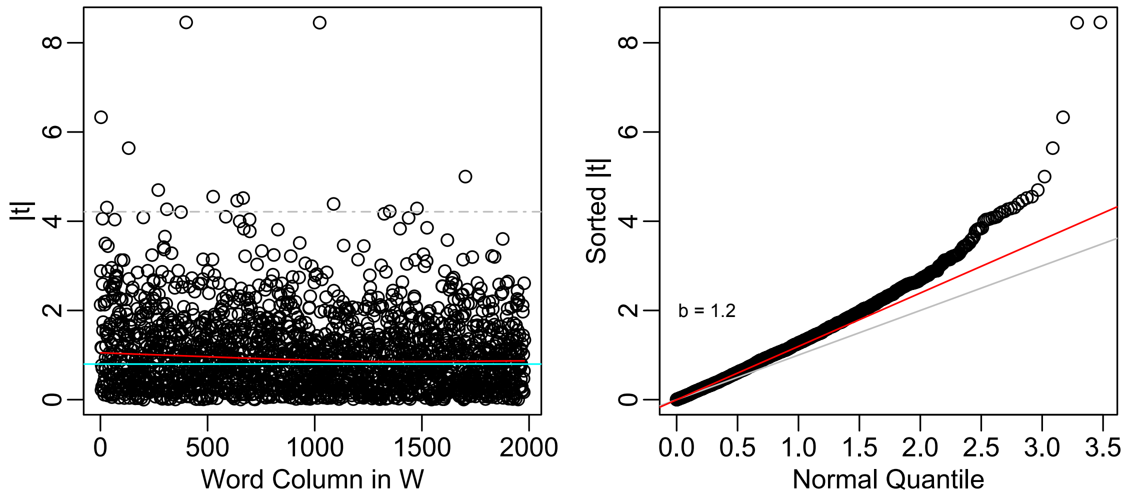
Each collection of singular vectors explains more variation than the simple model derived from several parsed words, but none find all of the signal captured by the larger collection of 2,000 word indicators. A regression of log prices on the 100 leading singular vectors attains $\overline{R}^2 = 0.49$, more than twice that of the model with parsed variables.

Table 2: *Multiple regression of log prices on counts from the document/word matrix W for the most common 2,000 words.* The table shows the 14 estimates that exceed the Bonferroni threshold for statistical significance.

	Estimate	Std. Error	t	Pr(> |t|)
vacant	-0.5518	0.0652	-8.46	0.0000
deed	-1.3155	0.1557	-8.45	0.0000
OOV	0.0373	0.0059	6.33	0.0000
units	0.1929	0.0342	5.64	0.0000
discount	-1.4959	0.2992	-5.00	0.0000
investment	-0.2334	0.0497	-4.70	0.0000
most	0.3350	0.0736	4.55	0.0000
bucktown	0.3570	0.0790	4.52	0.0000
sf	0.3305	0.0741	4.46	0.0000
pullman	-0.6244	0.1423	-4.39	0.0000
bedroom	-0.0978	0.0227	-4.31	0.0000
terraces	0.7074	0.1650	4.29	0.0000
fenced	-0.2466	0.0578	-4.27	0.0000
scb	-5.5914	1.3244	-4.22	0.0000

$s_e = 0.682$ with $R^2 = 0.766$, $\overline{R}^2 = 0.681$

Figure 5: *Distribution of the t-statistics for the regression of log prices on 2,000 word indicators (columns of W).*



Adding more singular vectors produces statistically significant, though diminishing improvements. The collection of 500 singular vectors of W produces $\overline{R}^2 = 0.61$, which is less than the $\overline{R}^2 = 0.68$ derived from individual word counts. Table 3 shows several of the estimated coefficients and summarizes the overall fits of these models. As often seen in principal components regression, the statistical significance of the singular vectors is not monotonic in the order of the singular vectors. That said, the leading singular vectors tend to be more relevant than those that follow. Figure 6 summarizes the significance of the estimated coefficients in the same fashion as Figure 5 with a plot of the absolute t-statistics and a half-normal plot. Most of the main leading singular vectors are significant, with an increasing proportion of insignificant variables as the position in the decomposition increases. In comparison to the significance for the coefficients of the word indicators, these singular vectors by-and-large are more consistently predictive with less noise. The slope of the fit in the half-normal plot (using the least significant 200 estimates) is 2.9. Residual analysis again finds fat tails with only a hint of heteroscedasticity.

Not only can one continue to add further singular vectors, one can also consider nonlinearities in the form of interactions among these singular vectors. The addition of interactions improves the fit of this model immensely by taking advantage of nonlinearities (*i.e.*, synergies among the eigenword structure). For example, a regression using just the first 20 singular vectors of W obtains $\overline{R}^2 = 0.31$. Adding interactions among these (an additional 190 explanatory variables since no powers are added) improves the fit significantly to $\overline{R}^2 = 0.41$. Fitting models with interactions drawn from a larger collection of features more generally requires some form of selection or regularization. We pursue this further in Section 5.

The third regression uses features derived from the SVD of the bigram matrix B defined in (2). For this analysis, we retained $k_B = 100, 200, \ldots, 500$ left and right singular vectors (k_B columns in each of U_B and V_B) and for each computed the associated matrix of correlations C. With $2\,k_B = 1,000$ left and right singular vectors, the largest $\overline{R}^2 = 0.66$, slightly more than the 0.61 obtained using the 500 singular vectors defined from W. Table 4 summarizes these fits. Using correlations from either the left or right singular vectors alone (derived from either U_B or V_B) explains significantly less variation; for example, a regression on 500 correlation vectors derived from U_B alone produces $\overline{R}^2 = 0.61$ (about the same as obtained from the 500 singular vectors of W). The use of the left and right singular vectors produces collinearity among the explanatory variables. A consequence of this collinearity is a large number of insignificant regressors in the fitted model. Figure 6(b) shows the p-values generated by the 500 singular vectors of correlations. Compared to the singular vectors derived from W shown in Figure 6(a), these regressors have more diffuse signal. The slope in the half-normal plot (again, derived from the least significant 200 estimates) is 2 (compared to 2.9 for the regressors derived from W). Also, the trend in the half-normal plot is concave rather than convex; the most significant variables are less significant than anticipated by the signal in the smaller estimates.

We can concentrate more of the regression signal into the leading components by using canonical correlation analysis. Figure 7 shows the canonical correlations between C_l and C_r. The correlations remain close to 1 for about the first 100 or so canonical variables. Rather than use the columns of C_l as regressors, we can use the canonical variables from this analysis. Figure 8 summarizes the estimates using the 500 predictors. The regression signal is now much more concentrated in the leading canonical variables, resembling the structure found by LSA (Figure **??**). The overall fit, of course, matches that obtained by using C_l since the canonical vectors are linear trans-

Table 3: *Multiple regression of log prices on singular vectors of the document/word count matrix* W. The first table shows estimated coefficients for first 10 and last 3 singular vectors with $k_w = 500$.

	Estimate	Std. Error	t value	Pr(>|t|)
D1	-62.0847	2.4891	-24.94	0.0000
D2	-7.6494	0.9373	-8.16	0.0000
D3	-12.5468	0.7558	-16.60	0.0000
D4	10.4948	0.8262	12.70	0.0000
D5	-0.8683	0.7645	-1.14	0.2561
D6	7.6848	0.8036	9.56	0.0000
D7	-17.8753	0.7547	-23.69	0.0000
D8	18.8346	0.7577	24.86	0.0000
D9	3.9515	0.7540	5.24	0.0000
D10	4.0974	0.7521	5.45	0.0000
⋮				
D498	1.2327	0.7516	1.64	0.1010
D499	1.3265	0.7516	1.76	0.0776
D500	2.5296	0.7516	3.37	0.0008

k_w	Residual SD	F	R^2	$\overline{R}^2$
100	0.863	72.1	0.496	0.489
200	0.810	45.9	0.561	0.549
300	0.778	35.5	0.601	0.584
400	0.765	28.5	0.620	0.598
500	0.752	24.3	0.638	0.612

Figure 6: *T-statistics of the singular value regressors for (a) the singular vectors of W and (b) the left and right singular vectors of B.*

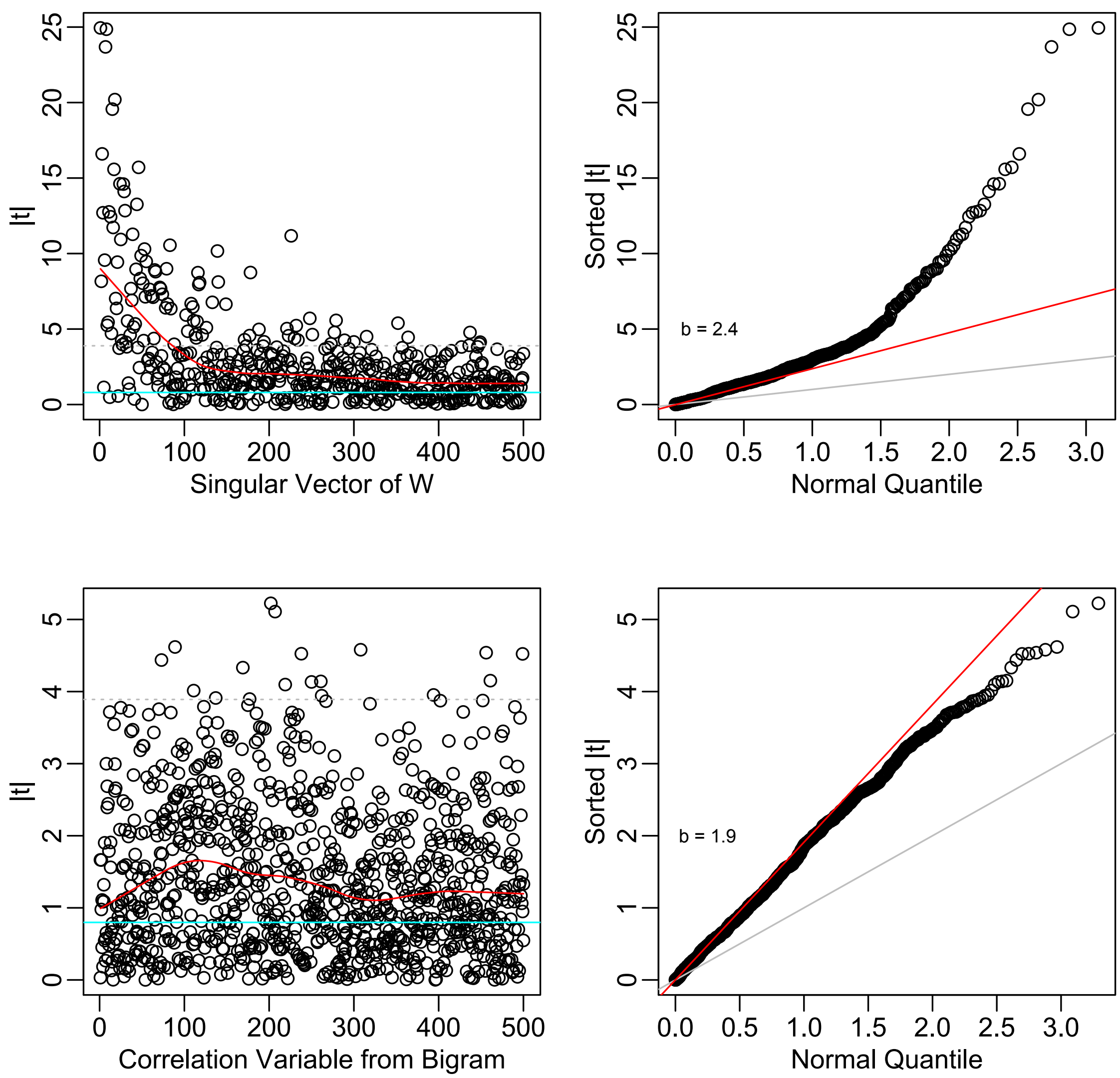
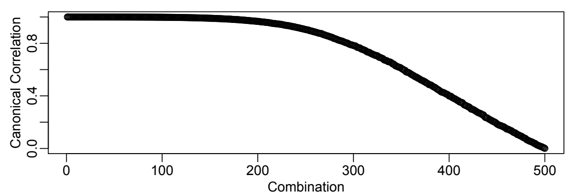


Table 4: *Multiple regression of log prices on regressors derived from the bigram matrix B.* Each regression uses correlations derived from k_B left and k_B right singular vectors.

$2\,k_B$	Residual SD	F	R^2	$\overline{R}^2$
200	0.842	40.0	0.527	0.514
400	0.779	26.8	0.605	0.583
600	0.750	20.6	0.645	0.614
800	0.724	17.4	0.679	0.640
1000	0.704	15.3	0.705	0.660

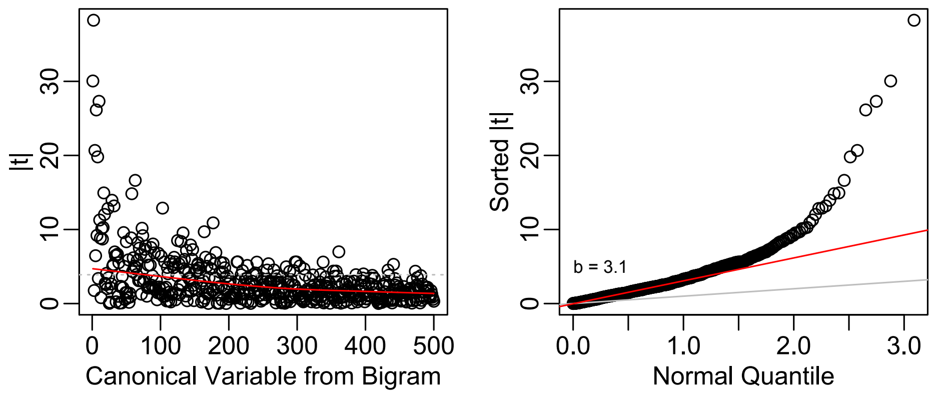
Figure 7: *Canonical correlations between the 500 columns of C_l and C_r.*



formations of C_l. We conjecture that CCA has this effect because, under the simple topic model introduced in Section 4 that follows, the left and right singular vectors of B measure essentially the same thing and aligning these via CCA produces a better measure of that common structure. This alignment, however, removes the distinction due to word order that the bigram matrix reveals. It may be the case that in this relatively small example (*i.e.*, relatively few documents) we lack enough text to exploit asymmetry in the use of words.

How well does combining both sets of variables perform? To find out, we start with the variables from the LSA, the singular vectors of W. The regression on the 500 singular vectors in U_W explains $\overline{R}^2 = 0.612$ of the variation in log prices. Adding the information from 500 more columns in C_l boosts the total to $\overline{R}^2 = 0.679$. Adding the remaining variation from C_r raises the total slightly (albeit significantly) to $\overline{R}^2 = 0.703$. Interestingly, the original parsed variables (Table 1) offer ever so slightly more predictive power. The improvement only adds 0.003 to $\overline{R}^2$, but this is highly significant (F=12.1).

Figure 8: *Canonical variables from the analysis of the dependence between C_l and C_r concentrate more signal into the leading regressors.*



4 Motivating Probability Models

To provide some explanation for the evident success of this direct approach to building regressors from text, we offer a hypothetical data generating process for text and study the implications of this DGP for regression modeling. The DGP is essentially that used in topic modeling (Blei, 2012). In machine learning, topic modeling is an unsupervised technique that clusters documents based on the presence of shared, underlying "topics" revealed by a hierarchical Bayesian model. Our method of featurizing text is also unsupervised, but we seek to predict an explicit response rather than uncover latent clusters. Nonetheless, we can study how our procedure would perform were there an underlying topic model.

4.1 Topic Models

We begin with the simplifying assumption that real estate properties possess varying amounts of K unobserved traits that influence both the value of a property as well as the language used to describe the property. For example, such traits might include the quality of construction, presence of renovations, proximity to desirable conveniences and so forth. In the context of topic models, these traits define the underlying topics shared by an ensemble of documents. In what follows, the subscript i indexes documents ($i = 1, \ldots, n$), m denotes words ($m = 1, \ldots, M$), and k indexes traits

($k = 1, \ldots, K$). Recall that words are tokens identified in the preprocessing of the text, not words in the usual sense. Let $y = (y_1, \ldots, y_n)'$ denote the column vector that holds the response that is to be modeled by regression analysis. In our application, y is the vector of the log of prices of real estate. (All vectors are column vectors.)

Within this model, traits influence the response via a familiar regression equation. The connection between traits and documents is given by an unobserved $n \times K$ matrix of latent features $\zeta = [\zeta_{ik}]$. Each row of ζ defines a mixture of traits that defines the distribution of words that appear in each document. To avoid further notation, we use subscripts to distinguish the rows and columns of ζ. The vector ζ_{i*} identifies the row of ζ associated with document i, and ζ_{*k} identifies the column of ζ associated with topic k:

$$\zeta = \begin{pmatrix} \zeta'_{1*} \\ \zeta'_{2*} \\ \vdots \\ \zeta'_{n*} \end{pmatrix} = (\zeta_{*1}\ \zeta_{*2}\ \cdots\ \zeta_{*K})\,.$$

ζ_{i*} specifies the distribution of traits present in the ith real-estate property; $0 \le \zeta_{ik} \le 1$ with $\sum_k \zeta_{ik} = 1$. We assume that the allocation of traits within each document is an independent realization of a Dirichlet random variable,

$$\zeta_{i*} \sim \mathrm{Dir}(K, \alpha_K)\,, \tag{4}$$

where α_m denotes the K-dimensional parameter vector of distribution. Given ζ, the K traits influence the response through a linear equation of the familiar form

$$\mathbb{E}\, y_i = \zeta'_{i*}\,\beta\,. \tag{5}$$

The coefficients β determine how the traits influence the response.

These traits also determine the distribution of words that appear in documents. This connection allows us to recover ζ — which is not observed — from the associated text. Assume that a trait defines a probability distribution over the word types in the vocabulary V. Let P_k denote the distribution of word-types used when describing trait k; in particular, P_{km} is the probability of using word type m when describing trait k. Our DGP models these distributions over word types as another set of independent Dirichlet random variables,

$$P_k \sim \mathrm{Dir}(M, \alpha_M), \tag{6}$$

where α_M is the M-dimensional parameter vector for the distribution. Collect these discrete distributions in the $K \times M$ matrix

$$P = \begin{pmatrix} P_1' \\ P_2' \\ \vdots \\ P_K' \end{pmatrix} . \tag{7}$$

The Dirichlet variables ζ and P together determine a distribution for the counts of words that appear in each document (its bag-of-words). First, we assume that the number of words in each document is another independent random variable, and for our simulation we use a negative binomial, formed by mixing Poisson distributions with parameters that have a Gamma distribution,

$$m_i|\lambda_i \sim \text{Poisson}(\lambda_i), \quad \lambda_i \sim \text{Gamma}(\alpha), \tag{8}$$

independently over documents. To 'construct' the ith document from this model, we sample m_i words from the underlying K topics by the following mechanism. Let w_{im} denote the mth word in the ith document. For this word, pick a topic at random (and independently) from the topic distribution identified by ζ_{i*}, say $k_{im} \sim \text{Multi}(\zeta_{i*})$. Then choose w_{im} from the multinomial distribution with these probabilities,

$$w_{im} \sim \text{Mult}(P_{k_{im}}), \quad i = 1, \ldots, m_i . \tag{9}$$

Hence, the vector of counts w_i for the ith document has a multinomial distribution whose parameters are determined by its mixture of traits:

$$w_i' \sim \text{Multi}(m_i, \zeta_i' P) \tag{10}$$

implying that $\mathbb{E}\, w_i'|m_i = m_i\, \zeta_i' P$.

Remark A. According to this DGP, the length of a document m_i does not affect the response; only the mixture of traits is relevant. Our results with real text summarized in the regression (Table 1) provide contradictory evidence: Document length has a significant impact on price.

Given that documents are generated by a topic model defined by equations (5) – (10), the challenge for making regression features from text is to recover the K-dimensional linear space spanned by ζ. The success of latent semantic analysis (LSA, principal components analysis of the word counts W) is particularly easy to see. Intuitively, LSA is well-matched to this DGP because both treat a document as a bag-of-words. Let D_m denote an $n \times n$ diagonal matrix with the word counts m_i along the diagonal. Then the expected value of the document/word matrix W is the sum of K outer products:

$$\mathbb{E}\, W = D_m \, \zeta \, P = D_m \sum_k \zeta_{*k} P_k' \,. \tag{11}$$

The expectation factors as an outer product, just as an SVD represents a matrix. That is, if we write $X = UDV'$, then we can express the matrix product as the sum $X = \sum_j d_{jj} u_j v_j'$ where u_j and v_j are the columns of U and V, respectively. For our models of text, the left singular vectors U_W from (1) are related to the allocation of traits over documents held in ζ. Of course, there are many ways to factor a matrix, and it is not apparent why the factorization provided by the SVD would be better than others. Our rationale relies on convenience (and the evident success in modeling real-estate prices), but one can argue that a decomposition that yields positive factors, namely non-negative matrix factorization NMF, would be more appropriate. Because both ζ and P are probability matrices, a constrained optimization that factors W into matrices whose rows are discrete probability distributions would be ideal. We do not explore these here.

Remark B. Expression (11) suggests that we should factor out D_m from W before doing the singular value decomposition so that variation of the lengths m_i does not contaminate the left singular vectors. This replacement of counts by proportions would then be followed by weighted least squares to down weight the influence of short documents. We explored several variations of this weighting, but none made a dramatic difference in the goodness of fit or out-of-sample accuracy.

The connection of this DGP to bigrams is less obvious and relies more on stronger assumptions. Bigrams count the frequency of the adjacent word types, a property we associate with the sequence of words rather than co-occurrence within a document. To see how the analysis of bigrams can nonetheless produce useful regressors, we need to add either stronger assumptions to our DGP or incorporate some type of sequential dependence. For example, we might assume that words associated with traits appear in *phrases.* As in the bag-of-words model, words within a phrase are drawn independently from the distribution defined by a trait, but the generating process samples within a trait for some length of time before transitioning to another trait (resembling an HMM). If these phrases are relatively long, then we can approximate the expected counts in the bigram matrix as a weighted outer product of the probability vectors for the traits. We can obtain the same heuristic in our DGP by assuming that the probability distributions P_k that define the traits have (nearly) singular support on words. That is, most words are associated with a unique trait (implying $P'_{k_1} P_{k_2} \approx 0$). In either case, the marginal probability of finding adjacent words types reveals the underlying probability distribution.

For instance, suppose the traits have disjoint support on words and that documents have common length $m_i \equiv m$. Then the probability of finding word types w_{m_1} and w_{m_2} from trait k adjacent to each other is

$$P(w_{m_1}, w_{m_2}) = \sum_i \left(\zeta_{ik}^2/n\right) P_{km_1} P_{km_2} = h_k P_{km_1} P_{km_2} \; . \tag{12}$$

Let $N = \sum m_i$ denote the total number of observed words. Using the expression (12), the expected value of the bigram matrix factors as

$$\tfrac{1}{N}\mathbb{E}\, B \approx \sum h_k P_k \, P'_k = P' H P \; , \qquad H = \mathrm{diag}(h_k) \; , \tag{13}$$

Again, a constrained factorization that produced probability vectors would be more appropriate if one truly believed this model. In an ideal world, the singular values of the SVD would capture the unobserved marginal probabilities h_k. Expression (13) also suggests why the left and right singular vectors of B should match, or at least define a common subspace.

The factorization of B defines the coordinates of eigenwords ($\mathbb{R}^M$). To obtain coordinates in document space ($\mathbb{R}^n$) for use in regression, we correlate the word counts in W with the eigenwords. For the ith description, if we pretend that the factorization of B is exact and approximate the word counts w_i by their expectation, then the first column of $w'_i U_b$ is

$$w'_i P_1 \approx m_i \zeta'_i P P_1 \; .$$

If the probability distributions of the traits are roughly singular as argued previously, then

$$w'_i P_k \approx m_i \zeta_{i1} P'_1 P_1 \; .$$

Hence, to this rough approximation, the correlation between w_i and the first left singular vector of B is

$$\text{corr}(w_i, U_{B1}) = \frac{\zeta_{i1}}{(\zeta_{i*}'\zeta_{i*})^{1/2}}$$

The L2 normalization is just right for canceling P_1, but leaves a constant factor $\|\zeta_i\|$.

Of course, even in expectation, the factorization of the bigram B will not match (13); the SVD only recovers a basis for the range of B. Thus, the singular vectors will mix the probability vectors. We can then think about $U_B = P'O$ for some orthonormal matrix O. That is, ideally the singular vectors span the correct space, but in a different basis so that we observe (again, in expectation)

$$w_i' U_b \approx (m_i \zeta_i' P)(P'O) = m_i \zeta_i \text{diag}(P_k' P_k) O$$

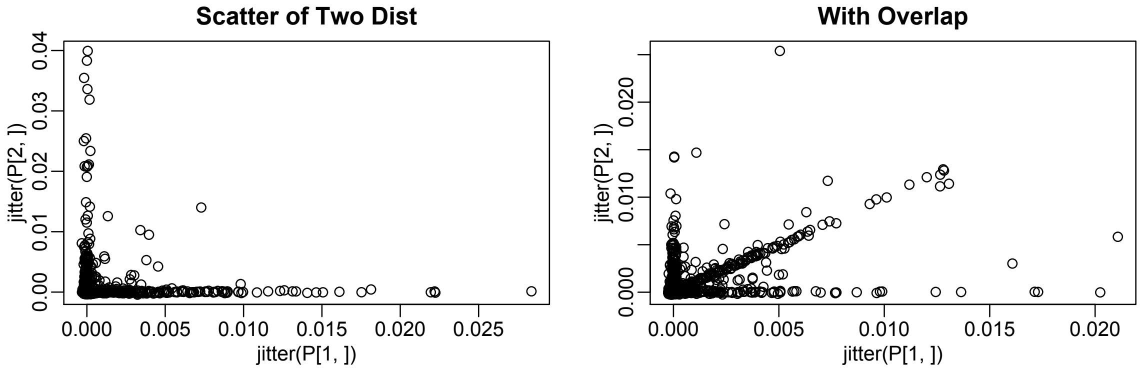
Hence, when computing the correlations between the observed counts w_i and the singular vectors, the norm of the probability distributions cancel and we obtain a rotation of ζ_{i*} vector. The rotation is the same, however, for all rows, and consequently our collection of regressors spans the same space as the unrotated ζ.

Obtaining a the relevant ζ-coordinates for a new document is routine in this case. One simply mimics the process used to identify/estimate ζ for the observed cases by correlating the counts for the new description, say w_{new}, with the matrix defined by the eigenwords.

4.2 Examples: Simulated Data from Topic Models

To get a better handle on how the singular value decomposition works, consider a simulated world in which a topic model generates text from a vocabulary of $M = 2,000$ words types. Assume that words in $n = 6,000$ documents are generated by mixture of $K = 10$ topics, with each topic defined by one of K distributions $P_1, \ldots, P_{10}$ over the 2,000 word types. Hence, p_1 lies in the M-dimensional simplex. Similarly, the distribution of topics within the ith document is distributed as $\zeta_{i*} \sim \text{Dirichlet}(K, \alpha_k = 0.1)$. The response for a document is a weighted sum of the mix of topics within the document. We consider two situations, first the idealized case in which topic distributions are essentially disjoint, and then with topic distributions share common words.

Figure 9: *Probabilities assigned to words by two simulated topic distributions with (a) nearly disjoint support and (b) with overlapping words.*



4.2.1 Nearly Disjoint Topic Distributions

For this example, we simulate P_k by independently drawing from a Dirichlet distribution with parameters M and $\alpha_m = 0.05$ for all $m = 1, \ldots, M$. Small values of α_m lead to 'spiky' distributions with little overlap. For example, Figure 9(a) graphs the probabilities assigned by two of the 10 distributions for this first simulation. The defining constants for this first example are:

$$M = 2000 \text{ word types}, \quad n = 4000 \text{ documents}, \quad K = 10 \text{ topics}$$

with random variables defined by

$$\begin{array}{rcl} m_i & \sim & \text{Pois}(\lambda_i), \lambda_i \sim \text{Gamma}(30, 1)\ , i = i, \ldots, n, \qquad \text{(Negative Binomial)} \\ P_k & \sim & \text{Dir}(M, \alpha_m = 0.05), \quad k = 1, \ldots, K, \\ \zeta_{i*} & \sim & \text{Dir}(K, \alpha_k = 0.10), \quad i = 1, \ldots, n, \\ k_{im} & \sim & \text{Mult}(\zeta_{i*}), \quad m = 1, \ldots, m_i, \\ w_{im}|k_{im} & \sim & \text{Mult}(P_{k_{im}}), \quad i = 1, \ldots, n\ , \\ y_i & \sim & N(\zeta'\beta, 0.5^2), \quad i = 1, \ldots, n\ . \end{array}$$

With these choices, $\overline{R}^2 \approx 0.92$ for the regression of y on ζ.

The distribution of words produced by this topic model loosely resembles the distribution found in real estate ads. Compare Figure 10(a) to Figure 2 from the real estate data. Though a good match for the less common words, the most common words are not as common as one might want. The common words do not have a high enough frequency (disjoint topics lack common words like 'a' and 'the' and punctuation as

Figure 10: *Distribution of word frequencies for simulated topic data, based on (a) disjoint word distributions and (b) over-lapping distributions.*

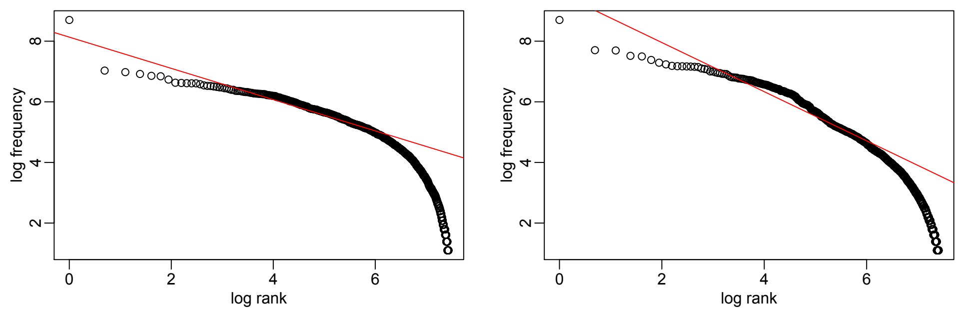


Table 5: *Fits to regression in simulated topic data using singular vectors of W and B for two topic distributions.*

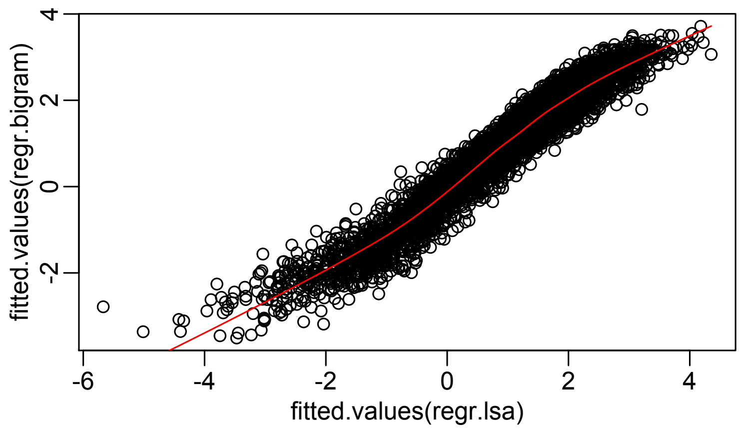
Topic Structure	Num Regressors	Origin	$\overline{R}^2$
Disjoint	100	W	0.746
	200	B	0.788
Overlapping	100	W	0.516
	200	B	0.626

found in the real-estate listings shown in Figure 2), and the frequencies for rare words drop off too quickly.

Table 5 summarizes the fits using $k_W = 100$ singular vectors and $k_B = 100$ left and right singular vectors of B. The fits recover much, but certainly not all of the underlying regression structure (for which $R^2 \approx 0.92$). The fitted models produce similar fitted values, as shown in Figure 11. The two fits, however, deviate for documents with either lower or higher values of y.

Because the topic distributions are nearly disjoint, one can identify the number of topics K from the singular value decompositions. The value of K is most apparent in a canonical correlation analysis of the singular vectors of B. Figure 12 graphs the canonical correlations for the left and right singular vectors of the bigram matrix for the simulated text. The clear break in the sequence of singular vectors confirms the

Figure 11: *Scatterplot of fitted values for the two regressions based on singular vectors of W and B.* The shown smooth curve is the lowess fit, and the correlation $r \approx 0.97$.



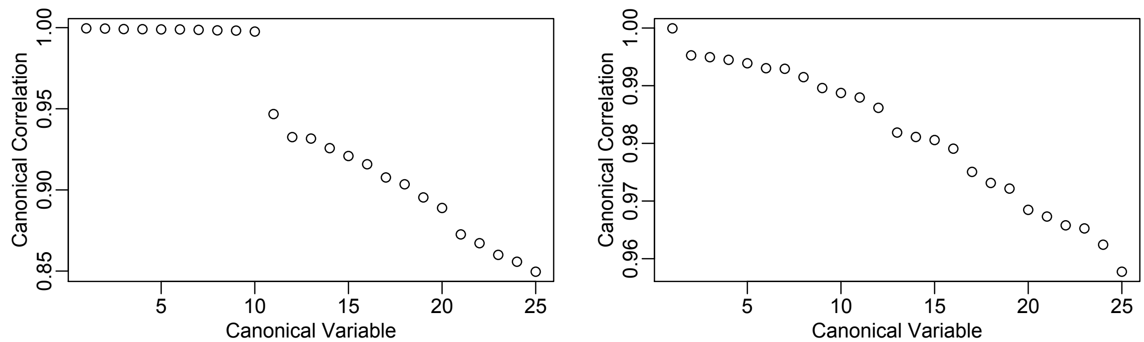
presence of $K = 10$ topics.

4.2.2 Overlapping Topic Distributions

The ability to recover the underlying regression structure is diminished in the presence of overlapping topics. The simulation in this section is virtually identical to the previous one, but for overlapping topics. The simulation again has a vocabulary of $M = 2,000$ words, $n = 6,000$ documents, and $K = 10$ topics. Rather than simulate the topics distributions independently, however, we introduce a common structure. The topic distributions share 150 “common words” that follow with probabilities from a Dirichlet distribution. Let $X \sim \text{Dir}(150, 2)$ and let ξ denote the vector obtained by concatenating M - 150 zeros onto X, $\xi' = (X, 0, \ldots, 0)$. The distribution that defines the kth trait is then $P_k = \xi/2 + Z/2$ with $Z \sim \text{Dir}(\alpha_m)$. Figure 9(b) plots the probabilities of two distributions generated in this manner. The diagonal points show the substantial overlap. The number of topics per document and the marginal distribution of words per document are both similar to the prior simulation and are not materially influenced by the introduction of this overlap.

The same cannot be said for the regressors produced from singular value decompositions. The lower portion of Table 5 summarizes these fits. The regression on 100 singular vectors from W now captures $\overline{R}^2 = 0.516$ of the variation in the response, and the regressors from the bigram obtain $\overline{R}^2 = 0.626$. The correlation between fitted

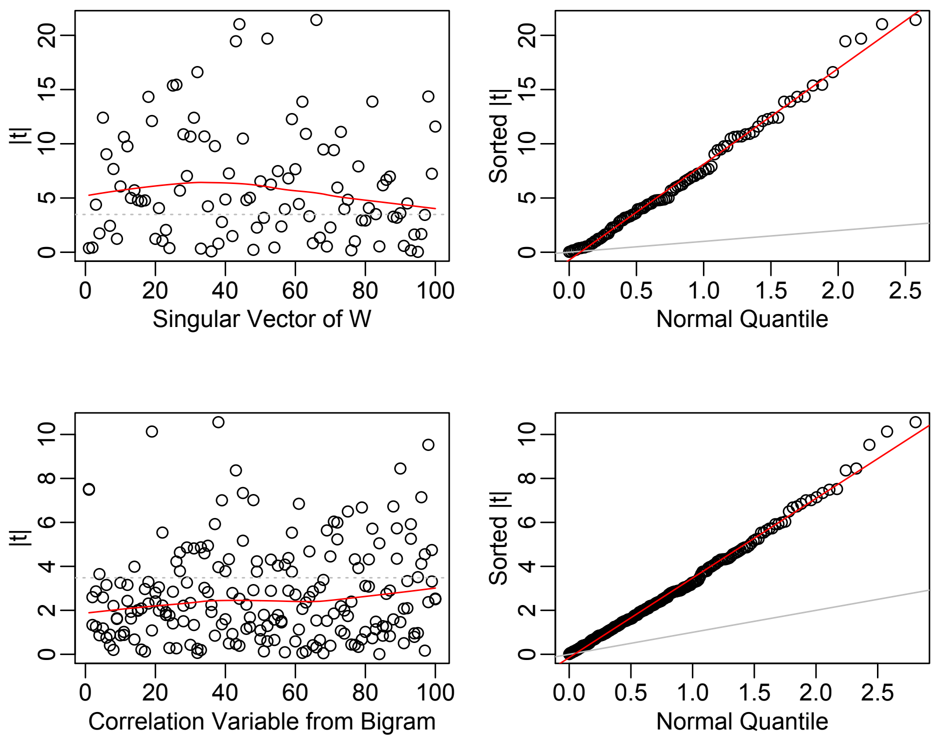
Figure 12: *Canonical correlations for left and right singular vectors of the bigram matrix B of simulated topic data.* (a) The underlying topics are nearly disjoint in support. (b) The underlying topics overlap.



values falls to 0.86 (down from 0.97 when no overlap was introduced). Curiously, the bigram is less influenced by the overlap. Canonical correlation of the singular vectors of B also no longer reveals the number of topics (Figure 12(b)). The canonical correlations between left and right singular vectors of B now provide no hint as to the choice of K.

Figure 13 summarizes the statistical significance of the coefficient estimates in this model. The distribution of significance is rather different from that seen in the real estate models (Figure 6). For example, statistical significance decays in the real estate model as one moves from the leading singular vectors of W down to smaller vectors. The simulated results show no hint of greater significance in the leading singular vectors and have a diffuse signal. Also, both half-normal plots are very nearly linear, with little of the "superstar" character seen in the t-statistics from the regression models for real estate. What is similar to the model for prices is that the signal is more diffuse in the regression using correlation regressors derived from B. The slope of the line fitted to the half-normal plot of the t-statistics for the singular vectors of W is 8.8, whereas that for the regressors derived from B is 3.6.

Figure 13: *Distribution of statistical significance in the regression with simulated, overlapping topic models.* Results first for the regression on singular vectors from W, then those derived from the bigram matrix B.



5 Variable Selection and Cross Validation

The previous examples routinely estimate regression models with hundreds of explanatory variables. Though large by traditional standards, these models do not suffer from the problems associated with over fitting because we have not used the data to pick the model. We simply fit a large collection of regressors. Evaluating such models is thus the province of classical statistical tests and criteria such as adjusted r-squared.

As evidence that these models are not over fit, we offer the following example. Consider the LSA regression that uses 500 principal components of W. Adjusting for the effects of estimating the 500 coefficients and the intercept anticipates the out-of-sample mean squared error of prediction to be $s_e^2(1 + (p+1)/n)$. This simple approximation averages over the regression design, ignoring leverage effects. Plugging in the unbiased estimate $s_e^2 = 0.5649$ gives $0.5649(1 + 501/7384) = 0.603$. Because the values of the regressors in the test data do not match those in the training data, this estimator is typically smaller than the observed MSE by an amount depending on variation in the regressors.

For comparison, we performed repeated 10-fold transductive cross validation. Transductive cross-validation presumes that the full collection of regressors is available for the training data, which in this case implies that we have all of the data available to perform the principal components analysis. Only the values of the response are hidden in each fold of the cross-validation; the collection of regressors is fixed. We repeated the 10-fold cross-validation 20 times, each time randomly partitioning the cases into 10 folds. The observed average squared error was slightly higher than anticipated at 0.614 ± 0.007, but basically agreed with the estimate from the fitted model.

6 Lighthouse Variables and Interpretation

Though regression models are seldom causal, one is often tempted to interpret properties of the solution within the domain of application. Because the predictors computed from the decompositions in the prior section describe subspaces rather than some specific property of words, interpretation is essentially non-existent.

To obtain features that are interpretable, we exploit the pre¡sence of characteristics that are occasionally observable. For example, most home listings do not include the number of bathrooms. An explanatory variable obtained by parsing this count is missing for 74% of the property listings. We can use this partially observed variable, however, to construct a more interpretable variable from either the principal components of W or the bigram variables.

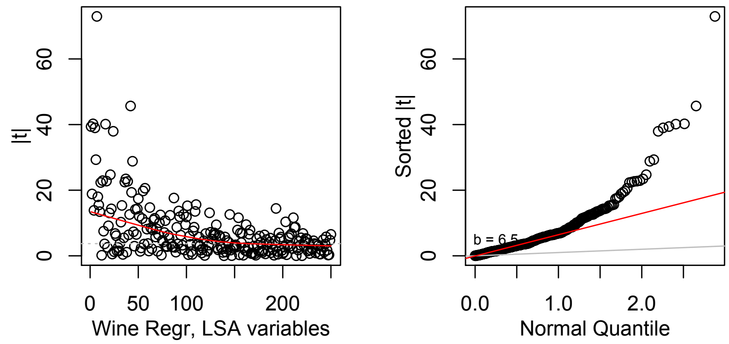
Let $z \in \mathbb{R}^n$ denote the partially observed or perhaps noisy data that measures a substantively interesting characteristic of the observations. For our example, z is the partially observed count of the number of bathrooms. Rather than use z directly as a predictor of y, we can use it to form an interpretable blend of, for instance, U_W. In particular, we simply regress z on these columns, finding the linear combination of these basis vectors most correlated with the observed variable. This variable, call it $\hat{z}$ then becomes another regressor. Because such variables can be used to guide the construction of interpretable combinations of the bases U_W and C, we call these lighthouse variables.

In our example, the correlation between the number of bathrooms and the price of the listing is 0.42 for listings that show this characteristic. This correlation is much smaller (0.19) if we fill the missing values with the mean number of bathrooms (Figure 4). If we form the projection $\hat{z}$ given by regressing the observed counts on the corresponding rows of U_W, this new regressor has correlation 0.29 with the log of price.

7 Summary and Next Steps

Our analysis here shows that one can exploit well-known methods of multivariate analysis to create regressors from unstructured text. Compared to iterative methods based on MCMC, the computations are essentially immediate. Surprisingly, the resulting regressors are quite predictive in several examples we have explored. For example, we used this same methodology to model ratings assigned to wines based on tasting notes. The tasting notes themselves are typically shorter than these real estate listings (averaging about 42 tokens compared to 72 for the listings), but we have a larger collection of about 21,000. Using the methods demonstrated here, a regression using 250 principal components of W explains about 66% of the variation in ratings, we a remarkably similar distribution of effect sizes as shown in Figure 14. Similarly, regression on the 250 left and 250 right regressors constructed from the bigram matrix explains about

Figure 14: *Regression t-statistics from a model that predicts wine ratings using 250 principal components of W based on 21,000 wine tasting notes.*



68% of the variation. We plan to explore other applications in the near future.

The connection to topic models is an important aspect of these results. Topic models define a DGP for which the regressors that we construct capture the underlying data-generating mechanism. If one accepts topic models as a reasonable working model of the semantics of text, then it is no accident that regressors constructed from text are predictive.

Our work here merely introduces these methods, and we hope that this introduction will encourage more statisticians to engage problems in modeling text. Our results here also suggest several directions for further research:

n-grams. Our example uses bigrams to capture word associations captured by adjacent placement. Other reasonable choices define different measures of context, such as trigrams (sequence of 3 words) or skipped bigrams (words separated by some count of tokens). Some preliminary results show, for instance, that trigrams offer modest gains, albeit at a nontrivial increase in computation.

Transfer learning. Transfer learning refers to learning what can be extrapolated from one situation to another. In our context, it would be of interest to learn how well models developed from data in June 2013 work when applied to data from later time periods or different locations. It is evident that the models shown here would not perform so well applied to the language of a different market, such as in Miami or Los Angeles. Not only do the characteristics of valuable properties change, but local conventions for phrasing listings are also likely to be different. Having a methodology for distinguishing idiosyncratic local features from those that generalize in time or space would be valuable.

Alternative forms of tokenization. Would be interesting to explore the use of stemming to reduce the number of word types and with a larger collection of documents, to explore annotation (that would distinguish words by their part of speech). Further parsing, lexical analysis. Some readers will be troubled by the simplicity of the bag-of-words representation of a document. Our methods understand neither the English language nor the rules of grammar and spelling. They do not attempt to untangle multiple uses of the same word. Linguists have debated the ability of such representations to reveal the meaning of language, and it is clear that the bag of words representation loses information. Just imagine cooking from"bag of words" recipe or following a "bag of words" driving directions. Nonetheless, this very direct representation produces very useful explanatory variables within our application. We leave open the opportunity to embellish this approach with more domain specific methods of parsing, such as adding part-of-speech tags and lexical information.

Use of unsupervised data. Most words are used only once or twice, meaning that we lack enough data to identify their connection to the response or indeed to other words. As a partial remedy, it may be possible to build regressors that represent such words from larger, more generic text such as the collection of n-grams collected by Google. Using this supplemental unsupervised data requires solving the problem of transfer learning, at least to some degree, but opens the door to much more extensive examples.

Variable selection. The distribution of effects (such as shown by the $|t|$ statistics of the text-derived regressors) are poorly matched to the so-called 'nearly black' model commonly adopted in research on the theory of variable selection. Rather than have most of the predictive power concentrated in a very small number of regressors, these regressors spread the power of many.

It would also be interesting to explore these models for nonlinearity. Variable selection is perhaps unnecessary for using the regressors derived from U_W and C, but essential if one hopes to detect and incorporate nonlinearity. In particular, searching for nonlinearity – such as interactions – requires variable selection. Even a very large corpus of documents looks small compared to the number of possible second-order interactions. Finally, the ordered presentation of the $|t|$ statistics suggests an opportunity for variable selection derived from alpha investing (Foster and Stine, 2008). Alpha investing is a procedure for testing a sequence of hypotheses that benefits from *a priori* ordering of the tests.

Acknowledgement

The authors thank Vanessa Villatoro from Trulia's PR Team for allowing us to scrape the data from their web site.

References

Baayen, R. H. (2002). *Word Frequency Distributions.* Springer, New York.

Blei, D. (2012). Probabilistic topic models. *Communications of the ACM* **55** 77–84.

Bullinaria, J. A. and Levy, J. P. (2007). Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior Research Methods* **39** 510–526.

Clauset, A., Shalizi, C. R. and Newman, M. E. J. (2009). Power-law distributions in empirical data. *SIAM Review* **51** 661–703.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K. and Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science* **41** 391–407.

Foster, D. P. and Stine, R. A. (2008). α-investing: a procedure for sequential control of expected false discoveries. *Journal of the Royal Statist. Soc., Ser. B* **70** 429–444.

Halko, N., Martinsson, P. and Tropp, J. A. (2010). Finding structure with randomness: stochastic algorithms for constructing approximate matrix decompositions. *Siam Review* **53** 217–288.

JURAFSKY, D. and MARTIN, J. H. (2009). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition (Second Edition).* Pearson.

LANDAUER, T. K. and DUMAIS, S. T. (1997). A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review* **104** 211–240.

MANNING, C. D. and SCHÜTZ, H. (1999). *Foundations of Statistical Natural Language Processing.* MIT Press, Boston, MA.

TABOADA, M., BROOKE, J., TOFILOSKI, M., VOLI, K. and STEDE, M. (2011). Lexicon-based methods for sentiment analysis. *Computational Linguistics* **37** 267–307.

TURNEY, P. and PANTEL, P. (2010). From frequency to meaning: vector space models of semantics. *Journal of Artificial Intelligence Research* **37** 141–188.

ZIPF, G. K. (1935). *The psycho-biology of language.* Houghton, Mifflin, Oxford, England.