Online Appendix – Disclosure Sentiment: Machine Learning vs Dictionary Methods

This online appendix includes the following sections:

1) Data preparation and machine-learning model assumptions
2) Description of random forest regression trees
3) Description of the earnings announcement press releases results

SECTION 1: Data Preparation and Machine Learning Model Assumptions

Textual Data Preparation

We use the following steps to clean our textual data in preparation for use in the machine learning algorithms:

1. Convert text to lower case.
2. Remove stop words (e.g., 'and', 'the', 'are', etc.).
3. Remove numbers and punctuation. Convert "%" to "percent".
4. Stem words. We use the Porter Stemmer algorithm available from the nltk.stem.porter Python library.
5. Create ngrams including unigrams (i.e., single words) and bigrams (two-word phrases). We create ngrams within sentences to avoid including words in an ngram that come from different sentences.

For example, consider the following sentences:

> *We are pleased with our results this quarter. Earnings increased by 15% relative to last year.*

We first convert the sentences to lower case and replace stop words with an underscore (i.e., "_"):

> *_ _ pleased _ _ results _ quarter. earnings increased _ 15% relative _ last year.*

We then split the sentences, remove punctuation, replace numbers with an underscore, and convert "%" to "percent":

> *_ _ pleased _ _ results _ quarter*
> *earnings increased _ _ percent relative _ last year*

We then use the Porter stemmer algorithm to stem each word:

> *_ _ pleas _ _ result _ quarter*
> *earn increas _ _ percent rel _ last year*

We then create unigrams and bigrams (ignoring "_"):

> *UNIGRAMS: earn, increas, last, percent, pleas, quarter, rel, result, year*
> *BIGRAMS: earn increas, last year, percent rel*

We then count the number of occurrences of each ngram within each disclosure.

In the machine learning models, we also remove sparse (i.e., infrequent) ngrams that are unlikely to have a significant impact on the machine learning algorithms. If included, these ngrams would dramatically increase computing time. We define these sparse ngrams as those that are used in fewer than 10 disclosures.

Support Vector Regression

We implement the support vector regression model using the SVMlight package developed by Cornell Computer Science Professor Thorsten Joachims (available for download here: https://www.cs.cornell.edu/people/tj/svm_light/). See the appendix in Frankel, Jennings, and Lee (2016)

for a thorough discussion of the support vector regression model including its assumptions and parameters. We follow the Cherkassky and Ma (2004) method for estimating both the epsilon and C parameters.

Random Forest Assumptions

We implement the random forest model using the RandomForestRegressor method from the scikit-learn package in Python. The RandomForestRegressor method requires two parameters chosen by the researcher. The first is the number of trees to use in the model (i.e., *n_estimators*), and the second is the number of features (i.e., ngrams) to use in each tree (i.e., *max_features*). The options for *max_features* are *sqrt*, *log2*, and *None*. The *sqrt* option uses the square root of the total number of features, the *log2* option uses the log (base 2) of the total number of features, and the *None* option uses all features in each tree. The default value in is the *sqrt* option. We use the default *max_features=sqrt* option and *n_estimators=5000* to train the random forest models.

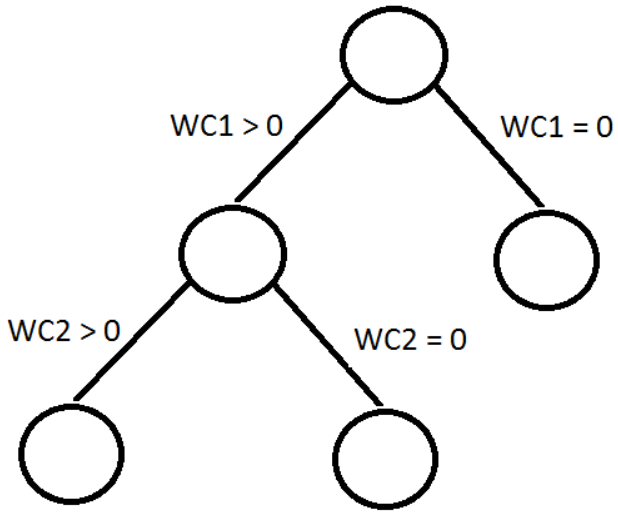
Supervised Latent Dirichlet Allocation Assumptions

We implement the supervised latent Dirichlet allocation method using the SLDA method from the slda package in Python (available for download here: https://github.com/Savvysherpa/slda; helpful implementation guidance available here: https://notebook.community/Savvysherpa/slda/examples/slda). The SLDA method requires the researcher choose the number of topics used in the algorithm. Dyer et al. (2017) use 150 topics when estimating unsupervised latent Dirichlet allocation, and Bao and Datta (2014) use 30 topics. We choose 200 topics to give sLDA even greater flexibility.

SECTION 2: Description of random forest regression trees

We describe regression tree estimation using a simple example. The first split of the data is based on the word count of word 1 (WC1). No further splits of the data minimize the SSE when WC1 is equal to 0, but when WC1 > 0, an additional split of the data based on the word count of word 2 (WC2) further minimizes the SSE within the resulting observations. The RF method is interactive in that it provides a different prediction for disclosures that have or do not have word 2 conditional upon having word 1. See a graphical representation of the estimation below.

Regression Tree Example



SECTION 3: Alternative Disclosures – Earnings Announcement Press Releases

As an additional test, we extend our analysis to earnings announcement press releases to examine whether our inferences extend to other disclosures. We apply the same procedure we use in estimating the dictionary-based sentiment measures and the machine learning models in the conference call setting to the earnings announcement press release setting. We identify 177,900 firm-quarter earnings announcement 8-K press releases between 2004 and 2019 with sufficient data to calculate the dependent and independent variables. We provide descriptive statistics for this sample in Panel A of Table OL1. The descriptive statistics are similar to those reported for the 10-K sample (Panel A of Table 1). Not surprisingly, the firm-quarter observations in the conference call sample (Panel B of Table 1) are larger, have higher growth, higher turnover, and higher institutional ownership.

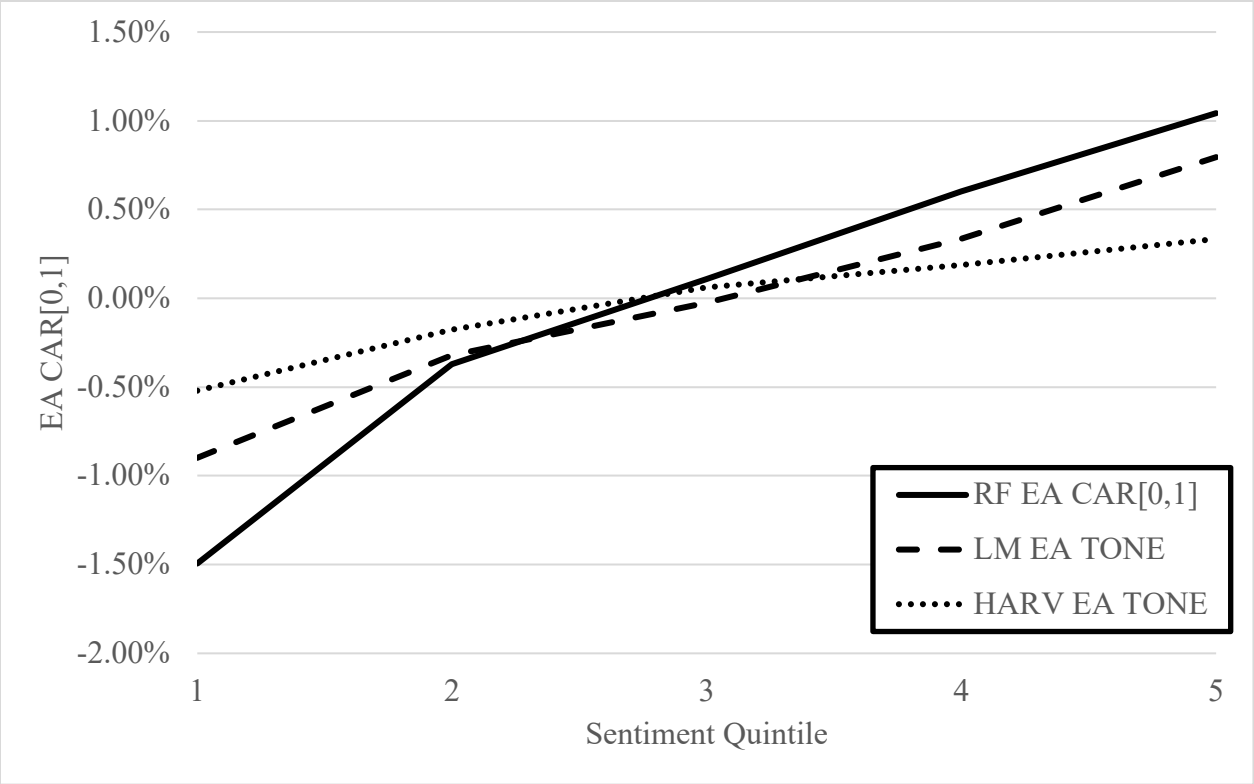
We next re-estimate Equation 2 using the earnings announcement press release sample and present the results in Panel B of Table OL1. Similar to our previous analyses, we focus on the sentiment measures based on the LM dictionary (*LM EA TONE*$_{i,q}$), Harvard dictionary (*HARVARD EA TONE*$_{i,q}$), and the RF model (*RF EA CAR[0,1]*$_{i,q}$). The coefficient on each of these measures is positive and significant, suggesting that the dictionary-based and machine-learning measures capture earnings announcement press release sentiment. The adjusted-R^2 when *RF EA CAR[0,1]*$_{i,q}$ is the independent variable is equal to 4.49% and is 6.37% higher than the adjusted-R^2 of 4.23% when *LM EA TONE*$_{i,q}$ is the independent variable. Using a Vuong likelihood ratio test, the adjusted-R^2s for all three measures are significantly different at the 1% level.

Similar to Figures 2 and 3, we graph the average two-day earnings announcement return (*EA CAR[0,1]*$_{i,q}$) by quintile of *LM EA TONE*$_{i,q}$, *HARV EA TONE*$_{i,q}$, and *RF EA CAR[0,1]*$_{i,q}$ in Figure OL1. A steeper line suggests that the measure better captures the information released in

the earnings announcement. The $RF\ EA\ CAR[0,1]_{i,q}$ measure yields the steepest line of the three measures, which provides graphical evidence that the RF method better captures earnings announcement sentiment relative to the dictionary-based methods.

Figure OL1

Earnings Press Release Returns: Comparison of Dictionary Methods to Random Forest



This figure plots the average value of *EA CAR[0,1]* for quintiles based on *RF EA CAR[0,1]*, *LM EA TONE*, and *HARV EA TONE*.

TABLE OL1

Sentiment Measures and Earnings Press Release Returns

This table includes 177,900 earnings press release observations from 2004 to 2019 and reports results of regressions where the dependent variable is equal to the cumulative abnormal return during the [0, +1] trading window surrounding the earnings press release date ($EA\ CAR[0,1]_{i,t}$). Panel A reports descriptive statistics, and Panel B reports regression results. Standard errors are clustered by firm. All variables are defined in the appendix. All continuous variables are winsorized at the 1% and 99% levels. *, **, and *** represent significance at the 10%, 5%, and 1% levels, respectively.

Panel A: Descriptive Statistics

Variable	Mean	Std. Dev.	Q1	Median	Q3
$EA\ CAR[0,1]_{i,q}$	0.000	0.088	-0.037	0.000	0.038
$LM\ EA\ TONE_{i,q}$	-0.102	0.310	-0.333	-0.134	0.094
$HARV\ EA\ TONE_{i,q}$	0.285	0.123	0.202	0.286	0.369
$RF\ EA\ CAR[0,1]_{i,q}$	0.000	0.012	-0.006	0.000	0.006
$EARN\ SURP_{i,q}$	-0.001	0.017	-0.001	0.000	0.002
$MVE_{i,q}$	5,612.950	14,714.190	339.734	1,105.200	3,680.200
$BTM_{i,q}$	0.540	0.458	0.251	0.451	0.730
$TURNOVER_{i,q}$	2.263	1.870	1.048	1.755	2.847
$PRE\_FFALPHA_{i,q}$	0.000	0.002	-0.001	0.000	0.001
$INSTOWN_{i,q}$	0.565	0.354	0.255	0.671	0.873
$NASDAQ_i$	0.510	0.500	0.000	1.000	1.000

Panel B: Regression Results

	[1]	[2]	[3]
Intercept	-0.001	-0.009***	0.002
	(-0.787)	(-5.436)	(1.531)
LM EA TONE$_{i,q}$	0.015***		
	(21.329)		
HARV EA TONE$_{i,q}$		0.018***	
		(10.535)	
RF EA CAR[0,1]$_{i,q}$			0.563***
			(25.381)
EARN SURP$_{i,q}$	1.002***	1.016***	0.997***
	(36.622)	(37.079)	(36.578)
ln(MVE)$_{i,q}$	-0.000**	-0.000	-0.001***
	(-2.092)	(-0.663)	(-5.816)
BTM$_{i,q}$	0.008***	0.007***	0.007***
	(11.089)	(9.760)	(10.219)
TURNOVER$_{i,q}$	-0.001***	-0.001***	-0.001***
	(-5.207)	(-5.499)	(-5.121)
PRE_FFALPHA$_{i,q}$	-0.162	-0.068	-0.697***
	(-0.821)	(-0.347)	(-3.488)
INSTOWN$_{i,q}$	0.007***	0.007***	0.005***
	(9.378)	(9.974)	(7.630)
NASDAQ$_{i}$	-0.002***	-0.001***	-0.001***
	(-3.175)	(-2.977)	(-2.974)
#OBS	177,900	177,900	177,900
Adjusted R^2	4.225%	4.026%	4.494%