The color of finance words[*]

Diego García[†] Xiaowen Hu[‡] Maximilian Rohrer[§]

June 14, 2022

Abstract

Our paper relies on stock price reactions to color words, in order to provide new dictionaries of positive and negative words in a finance context. We extend the machine learning algorithm of Taddy (2013), adding a cross-validation layer to avoid over-fitting. In head-to-head comparisons, our dictionaries outperform the standard bag-of-words approach (Loughran and McDonald, 2011) when predicting stock price movements out-of-sample. By comparing their composition, word-by-word, our method refines and expands the sentiment dictionaries in the literature. The breadth of our dictionaries and their ability to disambiguate words using bigrams both help to color finance discourse better.

JEL classification: D82, G14.

Keywords: measuring sentiment, machine learning, earnings calls, 10-Ks, WSJ.

[*] We thank Simona Abis (discussant), Will Cong, Tony Cookson, Gerard Hoberg (discussant), Byoung-Hyoun Hwang (discussant), David Stolin (discussant), Jim Martin and Chenhao Tan for comments on an early draft, as well as seminar participants at Indiana University, INSEAD, the CU Boulder CS-NLP lab, the CU Boulder Finance division, the FutFinInfo webinar, NHH Finance brown bag, the 2021 FMA conference, the 2021 SFS Cavalcade, the 2020 European Finance Association meetings, and the Michigan State University Fall 2019 conference. This work utilized the RMACC Summit supercomputer, which is supported by the National Science Foundation (awards ACI-1532235 and ACI-1532236), the University of Colorado Boulder, and Colorado State University. The Summit supercomputer is a joint effort of the University of Colorado Boulder and Colorado State University.

[†] Corresponding author, Diego García, University of Colorado Boulder, Email: diego.garcia@colorado.edu; Webpage: http://leeds-faculty.colorado.edu/garcia/

[‡] Xiaowen Hu, University of Colorado Boulder, Email: xiaowen.hu@colorado.edu; Webpage: https://www.colorado.edu/business/xiaowen-hu

[§] Maximilian Rohrer, Norwegian School of Economics, Email: maximilian.rohrer@nhh.no; Webpage: https://www.maxrohrer.com

1 Introduction

Since Tetlock (2007), the literature in Finance and Accounting studying different types of textual data has flourished.[1] The current state of the art to measure sentiment is to use a "bag-of-words" approach, counting words in dictionaries that are specialized to Finance and Accounting jargon, namely those developed by Loughran and McDonald (2011) (LM dictionaries). This approach has been criticized as potentially having low power in comparison to more sophisticated machine learning techniques (Gentzkow, Kelly, and Taddy, 2019). Our paper contributes to this debate by constructing new dictionaries using techniques from the natural language processing literature (NLP) in Computer Science, explicitly comparing their composition and predictive power relative to the LM dictionaries.

In essence, we ask the question of whether a dictionary constructed using stock price reactions as the "supervisor" can compete with humans codifying what are positive and negative words.[2] We validate both dictionaries measuring their ability to predict stock returns around earnings announcements.[3] The machine learning (ML) algorithm performs significantly better in out-of-sample tests than approaches based on the LM dictionaries. Our main contribution to the literature is to show how the ML algorithm achieves such improvements, providing new tools to measure soft information in financial and accounting disclosures.

Our paper focuses on the transcripts from the conference call(s) associated with a firm's earnings release ("earnings call"), arguably the most important regularly scheduled event in a firm's calendar. Frankel, Johnson, and Skinner (1999) argue these live calls have significantly more new information than other regularly scheduled events, like the actual filing of the annual 10-K statements.

We use the multinomial inverse regression model (MNIR) of Taddy (2013), a standard machine learning technique from the Computer Science literature, to build our new dictionaries. The main output from this algorithm is a set of loadings on *n*-grams that characterize their sentiment (both positive and negative). Our positive/negative *n*-gram dictionaries, which we refer to as ML dictionaries, include those *n*-grams associated with positive/negative loadings from the MNIR model. While we focus on the MNIR algorithm in Taddy (2013), the sufficient reduction

[1] See Loughran and McDonald (2016, 2020) for recent surveys.

[2] We will stick to the label "humans versus machines" following the narrative in Loughran and McDonald (2020), even while our interpretation is "humans versus stock prices". As most social scientists, we will loosely use the terms supervised/unsupervised and machine learning (Israel, Kelly, and Moskowitz, 2020). We use the term "sentiment" as in Tetlock (2007) and Taddy (2013), but we could have used the term "soft-information" or other synonyms: we are simply trying to measure the content, positive or negative, of a given piece of text.

[3] We will use the verb *predict*, and its declinations, in a purely statistical sense. All our regressions involve contemporaneous returns, i.e. we measure returns from the closing price prior to the event, to the closing price after the event.

ideas behind other machine learning algorithms in the literature are likely to produce similar (or better) results.[4]

One of our goals is to develop a new set of dictionaries that can measure sentiment over general English discourse dealing with business matters, based on stock price reactions to earnings calls. We take the output of our MNIR estimates and reduce its dimensionality by requiring sufficient stability across samples (across time/industry). Our final calibration yields a set of a few hundred unigrams and bigrams, which we consider one of the main outputs of our research agenda.[5] These "plain money English" dictionaries perform excellent relative to the LM dictionaries using samples of earnings calls, 10-K releases and WSJ articles, the three corpora we study in our paper.[6]

When working with unigrams, we show the ML algorithm uncovers new words that have predictive power, but it also allows us to refine the LM word lists. For example, we find that the term *issue(s)* is very negative whereas *momentum* is very positive (neither included in the LM dictionaries). The ML algorithm does not consider *against* to be a negative term, or *confident* to be positive (both included in the LM dictionaries). Our empirical exercise on unigrams both expands, suggesting new words, and refines, excluding words, the existing dictionaries. We emphasize that the set of new words is small (less than 100 terms), and that the ML algorithm excludes the majority of LM words (only 18/30 of the LM positive/negative words overlap with the ML dictionaries).

We also show the role that bigrams perform when summarizing text, as they help to disambiguate positive and negative words. To use some salient examples, we will be making a difference between *solid demand* and *soft demand*; between *best quarter* and *best estimate*. The ML algorithm labels bigrams that include *leverage* as extremely positive, which are unlikely to be classified by human coders as positive or negative.

To quantify the improvements brought by our approach, we note that a baseline specification of the stock price reaction to the earnings call event with controls has an R^2 of 1.7%, which the LM dictionaries (no overlap with ML) raise to 2.1%. Using the ML dictionaries (no overlap with LM) has an R^2 of 4.6%, whereas using bigrams it is 4.5%. In the univariate specifications, the overlap LM/ML dictionary has the largest R^2, at 5.4%, despite having the smallest number of terms. The multi-variate specifications all point to the ML dictionaries as the main drivers

[4] Rabinovich and Blei (2014) and Kelly, Manela, and Moreira (2018) improve and extend the original Taddy (2013) algorithm.

[5] The code and data that accompanies our paper allows researchers both to customize and change our calibrations. See Section 4.5.

[6] We use the term "plain" in the spirit of the "Plain English initiative" of the SEC. Our goal is to capture language that is general enough that it can be applied in different contexts/documents.

of stock returns, since the LM terms have no marginal statistical significance, loading with the wrong sign in several specifications.

We study the external validity of the new dictionaries, and existing ones, across 10-K releases and WSJ articles. We ask whether the ML dictionaries constructed using the earnings calls corpus can predict price reactions to 10-K filings and WSJ article publication. We find that the ML dictionaries generated using earnings calls are much more informative than the LM dictionaries in the context of 10-K releases. While the 10-K release is not a particularly important event, with most of the information disclosed during the earnings call that precedes it, the ML dictionaries load with the right sign in all specifications, whereas the LM dictionaries do not. Using WSJ articles as our corpus, we also find strong external validation for the ML dictionaries, although in this case the LM dictionaries still have some marginal predictability, albeit lower. It is worthwhile noticing that both the unigrams and the bigrams in the "plain money English" dictionaries have significant coverage and color across such distinct corpora.

Loughran and McDonald (2020) defend dictionaries developed by individual researchers selecting words, against algorithm based dictionaries, "humans versus machines." They write: "There is a hesitancy for researchers to define a word list because of this subjectivity. For this approach to be effective, the process must be transparent and the resulting lists should be reasonably exhaustive." We share both data, dictionaries and code from our research project, so the reader can reproduce every single word our algorithm picks.

The literature on textual analysis in Finance started by studying news media (Tetlock, 2007), mostly due to data availability and computing constraints existing at the time. Much interest has also been paid to annual statements: from analyzing sentiment (Loughran and McDonald, 2011), to industry (Hoberg and Phillips, 2016) and geographical classifications (García and Norli, 2012). Over the last decade a myriad of other sources of text has appeared, from the minutes of FOMC meetings (Hansen, McMahon, and Prat, 2018) to Internet message boards (Antweiler and Frank, 2004; Das and Chen, 2007) and Bloomberg news feeds (Fedyk, 2020), among others. We focus on the transcripts of earnings calls (Matsumoto, Pronk, and Roelofsen, 2011; Larcker and Zakolyukina, 2012; Bochkay, Chychyla, and Nanda, 2019; Fedyk, 2021), mostly for the high signal-to-noise ratio they provide, which is critical for machine learning applications.

Our paper contributes to the literature measuring sentiment, creating new dictionaries of both unigrams and bigrams using machine learning techniques applied to earnings calls. The Harvard-IV dictionaries used by Tetlock (2007) were the norm for a long time in the social sciences. Loughran and McDonald (2011) refined these dictionaries for accounting and finance documents, using annual statements (10-Ks).[7] Muslu, Radhakrishnan, Subramanyam, and Lim (2015) study forward-looking statements in 10-K filings, Cookson and Niessner (2020) create lists of words to describe investment styles, Baker, Bloom, and Davis (2016) do a similar exercise trying to measure political uncertainty, and many LDA papers also use some type of dictionary to give content to topics.[8]

Our research follows the supervised approach advocated by Kogan, Levin, Routledge, Sagi, and Smith (2009), and Manela and Moreira (2017), but instead of focusing on volatility,[9] we study first moments (sentiment). Jegadeesh and Wu (2013)'s analysis is similar in spirit, picking words using stock price reactions, but focusing on the Loughran and McDonald (2011) dictionaries, rather than allowing the data to pick *n*-grams from a larger set. More recently, Ke, Kelly, and Xiu (2019) and Cong, Liang, and Zhang (2020) use machine learning techniques in the context of corpora from the Dow Jones Newswires and the Wall Street Journal, which include many important events for firms, but not as salient as the earnings calls we study. While their empirical design is similar to ours, their focus is on unigrams.[10] Meursault, Liang, Routledge, and Scanlon (2021) also study earnings calls using machine learning techniques, focusing on the post earnings announcement drift, rather than the words chosen by the ML algorithm.

The rest of the paper is structured as follows. In Section 2 we discuss our data, and how we construct the dictionaries that form the core of the empirical exercise. In Section 3 we present our main results, where we compare the performance of the different dictionaries in the context of the stock price reactions to earnings calls, 10-K releases and the publication of WSJ articles. Section 4 studies dictionary breadth, the LM words in more detail, and discusses the disambiguation that our bigram representation achieves. The Appendix includes further details.

2 Measuring sentiment

In this section we first discuss the financial text corpora that we study in our paper, as well as different NLP techniques we implement to clean and organize our datasets. We then discuss the particular machine learning algorithm that we will use for the rest of the paper, and introduce

[7] The literature that uses the LM dictionaries spans many corpora, including 10-K statements (Feldman, Govindaraj, Livnat, and Segal, 2010), newspaper articles (García, 2013), IPO prospectuses (Hanley and Hoberg, 2012), press releases (Solomon, 2012), earnings calls (Chen, Nagar, and Schoenfeld, 2018), and more (Loughran and McDonald, 2016).

[8] The literature on LDA methods in financial economics has exploded in the last few years. For some examples see Hoberg and Phillips (2016), Hansen, McMahon, and Prat (2018), Bybee, Kelly, Manela, and Xiu (2019).

[9] In a similar vein, Glasserman and Mamaysky (2019) use 4-grams to measure "news unusualness" and predict volatility in the context of the banking sector during the 2008 financial crisis.

[10] Loughran and McDonald (2020) discuss the Ke, Kelly, and Xiu (2019) dictionaries at some length.

our method for constructing new dictionaries. We end the section by describing our empirical approach.

2.1 Data

We study three different types of textual corpora that have been the focus of previous studies: earnings calls (Frankel, Johnson, and Skinner, 1999), 10-K statements (Loughran and McDonald, 2011), and WSJ journal articles (Goldman, Gupta, and Israelsen, 2022). We note that these three corpora are clearly related: 10-K statements are typically released shortly after the earnings calls, and WSJ articles are often associated with such public disclosures. At the same time, they are quite different types of text: both in terms of their content (spoken language scripted with Q&A in the case of earnings calls; written language, with much legal jargon in the case of 10-K; to jounalist-style writing in the case of the WSJ), and their size (a few thousand words for earnings calls, a few hundred for each WSJ article, and very large for annual statements).

The dataset on quarterly earnings calls is constructed by merging two datasets. Our first data source are transcripts of earnings calls gathered from Seeking Alpha between 2006 and 2020. The second is the earnings calls transcripts as provided by Wall Street Horizons, which covers the period 2009–2020. The intersection of these two datasets over the overlapping period 2009–2020 is virtually the same as their union over the same time period, with identical word counts: we use both simply to have a longer time series.

We impose several data filters and data requirements, following Loughran and McDonald (2011) closely. We require that the firm hosting the conference call can be matched to CRSP and Compustat[11] and that regression variables are available (see the Appendix for details). We also require firms to have at least 60 days with available trading volume and return in the year before and after the call date. We limit the sample to firms listed on NYSE, Nasdaq, and AMEX, that are reported on CRSP as ordinary common equity firms (share code 10 and 11), and that have a share price of more than $3 on the day before the call. Lastly, we exclude calls that have transcripts with less than 100 words. These selection criteria yield a sample of 85,530 events, associated with 3,229 unique firms.

We study the full text of the 10-K statements, as provided in Bill McDonald's webpage.[12] Our focus will be in predicting the stock market reaction over the four days around the release of the

[11] Matching is based on a combination of ticker and quarterly earnings release date (Compustat item RDQ).

[12] See `https://sraf.nd.edu/data/stage-one-10-x-parse-data/`. An earlier version of the paper studied only the management discussion and analysis (MD&A) section of the 10-K statement, which has been the focus of much of the literature (see for example Hoberg and Lewis, 2017, for a recent contribution), with similar results to those reported in this draft. Loughran and McDonald (2011) use both the full 10-K statement, and also the MD&A section.

10-K, mimicking our previous analysis using earnings calls and that in Loughran and McDonald (2011).

The dataset contains all annual reports (10-K) filed in the period 1996–2018 that can be matched to the CRSP database. We follow the sample selection in Loughran and McDonald (2011) considering stocks listed on the NYSE, Amex, or NASDAQ. We limit to all filings with available regression variables (size, book-to-market, share turnover, pre-filing period three factor alpha, filing period excess return, and Nasdaq dummy). We exclude firms with a stock price on the day before the call of \$3 or less, and require the firm to have at least 60 days of trading in the year before and after the filing date. We exclude filings with less than 2,000 words. Lastly, we include only filings with 180 days between them and only one 10-K filing per year and firm. The final sample includes a total of 76,922 observations.

We collect Wall Street Journal (WSJ) articles using Factiva, following the protocols in Goldman, Gupta, and Israelsen (2022).[13] We manually download all articles tagged in Factiva as associated with a given firm, starting with the list of firms ranked by frequency in the Ravenpack database. We limit our analysis to articles that mention at most seven entities, and that have a minimum of twenty words (after applying the NLP cleaning procedure detailed in the Appendix). Our manual collection results in a set of 144,383 unique articles associated with 189 unique firms. Since we use daily stock returns in our analysis, we merge all news associated with the same firm in a given day, and construct sentiment scores on this aggregated text. Our final dataset has 87,198 unique firm days.

Our paper will focus on the corpus from earnings calls, namely the transcripts from the call between the firm's management and analysts/investors, in order to construct the new ML dictionaries. The main reason for focusing on this corpus is that the signal-to-noise ratio of the earnings calls is significantly stronger than most other corporate events, i.e. relative to the release of the actual 10-K statements (Loughran and McDonald, 2011), which are typically filed after the earnings calls.[14] The essence of our approach relies on using stock price reactions to label *n*-grams as positive or negative: the machine learning algorithm is supervised by market reactions while trained. Having a strong signal-to-noise ratio in our empirical exercise is therefore critical.

To support our choice of earnings calls, we follow Griffin (2003) and compute the absolute excess return for each day around the three different events, normalized by its mean and standard deviation (computed in the period of -60 to -2 day around the earnings call date). Figure 1

[13] We thank Ryan Israelsen for sharing the details on the manual procedure that allows for downloading the WSJ articles from Factiva.

[14] See Li and Ramesh (2009).

plots the averages for the 10 days before and after the event for earnings calls (blue circles), 10-K releases (red crosses), and WSJ article publications (green triangles). We note that the average absolute value, under normality, should be around $\sqrt{2/\pi} \approx 0.8$ (dashed line).

Figure 1 shows that earnings calls are associated with significantly more volatile stock prices than both 10-K statements releases and the publication of WSJ articles. The average absolute excess returns on the earnings call event date is around 2.2 on the day of the event, and 1.8 the day after, both more than twice as large as the unconditional mean (0.8). We remark that the result on the day after the event is driven by the fact that many earnings calls are held in the afternoon, and we are measuring returns using closing prices. This also motivates our choice of a four day window around the earnings event to be inclusive regarding its associated stock price reaction.[15]

Figure 1 shows that there are stock price reactions associated with the 10-K release event, as well as the publication of articles in the WSJ, but the effect is much more muted. The effect, as in the case of the earnings call, lasts into the next day, with smaller magnitudes.[16] The smaller reaction to 10-K releases should not be surprising, as the earnings calls often happen a week before the formal submission/acceptance of the 10-K statement by the SEC. Similarly, coverage in the WSJ will typically lag the press releases of the earnings calls, and/or cover less important events (i.e. press releases associated with 8-K statements, industry news). This evidence argues that earnings calls are a better event for performing the type of supervised learning algorithm we implement in our paper.

2.2 Robust multinomial inverse regression

In this section we describe our main textual analysis tool, the multinomial inverse regression (MNIR) model of Gentzkow, Kelly, and Taddy (2019), as well as the extension we use to avoid over-fitting, which we refer to as "robust MNIR."

Our textual corpus is a set of n documents, i.e. the transcript from an earnings call T_j. We want to associate such text with the stock market reaction to the event, which we will denote by R_j. While the representation of T_j can be kept fairly abstract, for our purposes it will be a document-term-matrix (dtm) where we keep count of what terms (tokens), out of a set of p total n-grams, appear in each of the documents in the corpus of interest. We choose the most frequent p n-grams in the whole corpora to construct our dtm, using $p = 2^{14} = 16,384$

[15] Our results are identical with shorter windows.

[16] We note that when the 10-K is released on the same day or the day before the earnings call, which occurs in less than 20% of cases, the average absolute excess return is significantly higher than when 10-K is released two or more days after the earnings call.

(16K) in our baseline specifications for unigrams, and $p = 2^{16} = 65,536$ (65K) in our baseline specifications for bigrams. We discuss reducing/enlarging the dtm in Section 4.1.

The above dtm representation is a standard NLP approach to summarizing text, where the underlying document is represented by a sparse matrix. We note that there is some loss of generality, as we do not keep track of the sequence of words in the document. At the same time, using bigrams we are keeping some context, which will prove crucial when disambiguating words.

The MNIR model has a Bayesian flavor, belonging to a class of algorithms close to topic models (such as LDA).[17] The MNIR uses the conditional distribution of text given sentiment to obtain low-dimensional scores that summarize the information relevant for the stock return reaction. This is actually at the heart of many of these algorithms, where the Bayesian structure allows for considering both $R_j|T_j$ and $T_j|R_j$. The MNIR algorithm uses a lasso-style penalty on the first set of inverse regressions to construct a sufficient statistic Z_j, which can then be used for out-of-sample prediction.

The inverse regression of interest is stock returns onto word counts, which within a Bayesian framework with a given set of priors generates a set of posteriors on the influence of tokens (*n*-grams) on stock prices. It is important to note that in contrast to other methods, such as in Meursault, Liang, Routledge, and Scanlon (2021), we do not need to discretize our outcome variable, stock returns, as the MNIR model allows for continuous variables.

The MNIR model involves regressions of stock price reactions on individual *n*-gram counts, so it is related in spirit to the algorithm in Jegadeesh and Wu (2013), with two important differences: (i) the MNIR's inverse regressions are not joint regressions, which breaks the curse of dimensionality in typical machine learning fashion,[18] (ii) the lasso ($\mathscr{L}^1$) penalty and the MNIR's Bayesian structure yield different fits/estimates of the sentiment of *n*-grams.

For our purposes, the main output from the MNIR that we will explore is the loadings on each of the p *n*-grams that the algorithm generates.[19] These loadings are roughly evenly distributed into positive/neutral/negative in our baseline specifications. Thus, the MNIR algorithm allows us to classify the *n*-grams into two dictionaries: one consisting of *n*-grams with positive loadings, one consisting of those that have negative loadings.

[17] The discussion in Gentzkow, Kelly, and Taddy (2019), in particular Section 3.2, links the MNIR to topic models (see also the discussion in Rabinovich and Blei, 2014; Roberts, Stewart, Tingley, Airoldi, et al., 2013).

[18] The joint estimation advocated in Jegadeesh and Wu (2013) would be unfeasible with the number of *n*-grams that we consider, which is larger than the number of observations (earnings calls, 10K statements, WSJ articles).

[19] We highlight that our results are not sensitive to the choices of lasso penalties and set of priors that need to be specified for the estimation of the MNIR model. A higher lasso penalty will reduce the size of our dictionaries, as more *n*-grams end up with zero loadings, but our predictability results are robust to different parameterizations.

We note that when generating these dictionaries we are ignoring the size of the coefficients in the estimated MNIR model. We construct the positive/negative dictionaries in order to be able to compare the machine learning algorithm on the same terms as the standard bag-of-words approach, at the cost of penalizing the machine learning performance by ignoring the information embedded in the size of the estimated coefficients. One can consider this step an extra dimension reduction step in our algorithm, with a similar flavor to a lasso penalty, simplifying the final sentiment representation.

The choice of the MNIR algorithm, versus others in the literature, is motivated by its performance. Section 5.1 in Taddy (2013) shows that (1) MNIR is very robust to changes in parameter specifications, (2) compared to other leading textual analysis methods MNIR provides higher quality predictions with lower run-times.[20] We conjecture that using more modern methods will only widen the "machines versus humans" divide we document using the MNIR algorithm.

As most machine learning methods, the MNIR will overfit, i.e. pick too many terms that happen to be correlated with returns on the given training sample chosen. In order to mitigate its tendency for overfitting, we consider an extra convolution layer, which we label "robust MNIR." In particular, starting with a training sample with m observations, we will fit the MNIR to k different subsamples of size q. We will use $q = 5,000$ and $k = 500$ in our baseline specifications. We choose each of the k samples randomly, bootstrapping without replacement from the m events in the training sample.[21] For each different MNIR fit, we assign a positive/neutral/negative score (1/0/-1) for each of the p n-grams in our dtm. We can then ask about how many times a given n-gram has positive/negative/neutral loadings, and look for consistency across the samples to avoid overfitting.

The basic idea of this extra step is to penalize n-grams that are rare, but spuriously correlated with stock returns in the training sample. Each bootstrapped sample has a different mix of industries, time periods and firms, so requiring consistency across a large set of subsamples gets at the goal of measuring "plain money English." By requiring consistency across multiple subsamples in the training set we will significantly avoid overfitting. In our baseline specifications, the top 20 n-grams consistently appear 99%+ of the time as positive/negative across the k different bootstrapped samples.

[20] Section 5.1 in Taddy (2013) studies speeches from the 109th US congress and we8there restaurant reviews. MNIR is compared to text-specific LDA (both supervised and standard topic models), lasso penalized linear and binary regression, first-direction PLS, and support vector machines.

[21] A previous version of the paper performed this robustness step using 5x5 subsets across time and Fama-French industries. The results are very similar to those reported in this draft. Using randomization yields more subsamples, which makes the ranking of n-grams easier.

To summarize the "robust MNIR" algorithm, we will fit the MNIR model to $k = 500$ different subsets of our training sample. This will generate a different sentiment score for each of the p n-grams: the difference of the times they are scored as positive minus negative (as a percentage), which we will denote by D^+. For notational simplicity, we define $D^- = -D^+$, the difference of negative and positive scores in the 500 cross-validation subsets. Our final set of ML positive (negative) dictionaries will consist of those n-grams whose D^+ (D^-) score above a given cutoff. We set the cutoff for unigrams at 0.80, and that for bigrams at 0.45. As we will see in our empirical results, these are quite stringent criteria, which will result in only a handful of n-grams. This is at the heart of our robust MNIR algorithm: avoid misclassifications by requiring consistency across the bulk of the training sample. We will label the final dictionaries, constructed using the above algorithm as ML "plain money English" dictionaries, or simply ML dictionaries.

2.3 Creating sentiment scores

The standard approach to measure sentiment in the Finance literature is to start with a "bag-of-words", a collection of tokens that are labelled positive/negative by researchers. For example, Tetlock (2007) uses the Harvard-IV dictionaries, which were developed by psychologists, and consist of 1,637 positive words and 2,005 negative words. The dictionaries from Loughran and McDonald (2011) are a refinement of the Harvard-IV dictionaries, and include 347 positive and 2,345 negative terms.[22]

Once these bag-of-words are decided upon, a sentiment score is assigned using either the sum of the term frequencies of the members of each dictionary (normalized by the size of the document), or some variation that accounts for the incidence of a term across the corpus (tf-idf scores). We will implement our main analysis using term frequency weights throughout the paper.[23]

We represent a document j as a sparse vector $\text{tf}_j = [\text{tf}_{1j}, \ldots, \text{tf}_{pj}]$ of term frequencies for each of p tokens in a vocabulary $\mathscr{V}$. The term token is used to denote n-grams, consecutive combinations of n words. This is a standard approach in the NLP literature: summarize a document by the counts of tokens used in it as a document-term-matrix, where the rows represent the documents, and the columns represent the terms in a given dictionary. As discussed previously, the vocabulary $\mathscr{V}$ will consist of the p most frequent n-grams (16K for unigrams, 65K for

[22] We use the version of the Loughran and McDonald (2011) dictionaries as shared by the authors in their webpage as of 2022.

[23] An earlier draft showed tf-idf adjustments favor the ML algorithm versus LM, but they introduce slight challenges to the empirical exercise. In particular, we note that by construction, word counts are skewed to the right, since they are censored at zero on the left. The idf adjustment makes this skewness more pronounced.

bigrams).

A (positive/negative) dictionary is a subset of words from the p words in a given dtm, $\mathscr{D}_i$, i.e. a subset of the vocabulary $\mathscr{V}$. We can represent this as matrix D_i of the same row dimension as the dtm under consideration, with each column referring to each of the terms included in the dictionary.

We will define the sentiment for a given document j, and a dictionary of m words (positive/negative), as

$$S_j = \sum_{i \in \mathscr{D}_i} \left(\frac{\mathrm{tf}_{ij}}{N_j} \right), \tag{1}$$

where N_j is the total number of words in document j, and the index i runs through the words in the given dictionary $\mathscr{D}_i$.

In the case of unigrams our approach mimics that in the standard bag-of-words (Loughran and McDonald, 2011), in the sense that we start with a set of potential tokens, and we will assign to each of them a positive/negative/neutral sentiment score. Thus, we can directly compare our dictionaries to those in the literature. But our approach is broader in scope, as allowing for bigrams we can capture more nuanced aspects of the English language. We note that we do not impose term frequency limits on the corpus when computing LM scores, only when training the ML algorithm (where we use the dtms described above).

This construction of a sentiment score can use a dictionary of combinations of n-grams instead of just words, to the extent that we have a document-term-matrix in the right n-gram space, and a method that labels the different n-grams. Starting with a dictionary of an arbitrary size, the output from the MNIR algorithm allows us to create such a classification: those n-grams that get positive/negative loadings in the estimation of the sufficient reduction statistic.

To summarize, in what follows we will compute sentiment scores for each document in our corpus using the standard LM approach, and using the ML dictionaries as well. Since the latter can be constructed using unigrams or bigrams, we will have different sets of dictionaries developed by the machine learning algorithm. When dealing with unigrams, we will separate those terms that are included in both the LM and ML lists, and add those that are uniquely in the LM and ML lists separately. This allows us to determine the marginal contribution of each of the dictionaries in predicting stock price movements. In particular, we will look at the impact of the terms the ML algorithm agrees with the human classifications (LM), and what the marginal contributions of new terms may be.

2.4 Empirical design

Our main empirical approach is to study regressions of the form

$$R_{jt} = \beta S_{jt} + \gamma X_{jt} + \varepsilon_{jt}, \tag{2}$$

where t is the date of the event (earnings call, 10K release, WSJ publication); R_{jt} is the firm's buy-and-hold stock return minus the CRSP value-weighted buy-and-hold market index return over the 4-day event window (from close at $t-1$ to close at $t+2$), expressed as a percent; S_{jt} is one (or more) of our measures of sentiment, and X_{jt} are controls. We winsorize the sentiment variables, the controls, and returns at their 1/99% percentiles. See the Appendix for details on the variable definitions.

The main coefficient(s) of interest are measured by β. Throughout our analysis we normalize the sentiment scores S_{jt} to have unit variance, so the β coefficients can be interpreted as the marginal response of stock returns to a one standard deviation change in sentiment. We will use the magnitude and statistical significance of the coefficients as one of our comparison metrics, together with goodness-of-fit measures (adjusted R^2).

The specification in (2) is a standard event study with an unbalanced panel. We note that there is some clustering in the time dimension, which Loughran and McDonald (2011) deal with using Fama and MacBeth (1973) regressions. For simplicity, and since it is also standard practice for this type of empirical study, we keep the event-study structure and add both time (quarter-year) and industry fixed effects (FF49).[24] These controls complement the inclusion of hard data, in particular the standardized unexpected earnings (SUE), as well as lagged stock market returns, firm size, the book-to-market ratio, share turnover and a NASDAQ dummy (following Loughran and McDonald, 2011). We report standard errors clustered on FF49 industries and fiscal quarters. See the Appendix for details on the controls across our different corpora.

Since the dictionaries discussed in Section 2.2 are constructed in sample, we need to use standard cross-validation techniques for out-of-sample (OOS) verification. For simplicity, we use as a training sample all the events prior to a fixed date, and as the out-of-sample dataset all events after. Our algorithm first constructs the ML dictionaries using the training sample, and then creates the sentiment metrics and estimate the model (2) on the sample that we did not use for training. We remark that the particular sampling mechanism is not critical for our results. We could sample particular time periods, or do 80/20 training/validation, and our qualitative

[24] In a previous draft we implemented Fama-MacBeth regressions to complement our panel approach. Our results were qualitatively similar and quantitatively stronger. We note that earnings calls are more evenly distributed across the year than 10-K releases, which makes the benefits of Fama-MacBeth more muted.

and quantitative results are very similar.

3 Returns and text

Using the dictionaries from the machine learning algorithm described in Section 2, using 2006–2014 as the training sample, we study in Section 3.1 whether such classification has bite for predicting stock price reactions out-of-sample (2015–2020). In particular, we compare the performance of the machine learning algorithm to that from the standard bag-of-words approach (Loughran and McDonald, 2011). In Section 3.2, we take the dictionaries constructed from the earnings calls, using the full sample (2006–2020), and see if they have external validity, i.e. whether they can predict stock price reactions to annual statements (10-K) filings and/or the publication of articles regarding a firm in the Wall Street Journal.

3.1 Human versus machine dictionaries

In this section we present horserace regressions between sentiment metrics constructed using the machine learning algorithm described in the previous sections, and those constructed using the dictionaries from Loughran and McDonald (2011). Our empirical approach is rather simple: we compare the predictability in specifications as in (2) when the sentiment variable is constructed using different dictionaries.

We start by comparing the dictionaries generated by the ML algorithm developed in Section 2, fitted using the earnings calls corpus from 2006–2014, to the standard LM dictionaries. We choose the n-grams using the robust MNIR steps from Section 2.2, setting the criteria for inclusion at 80% for unigrams, and 45% for bigrams. We discuss these thresholds at more length in Section 4.1.[25]

We then build the dictionaries and sentiment scores as outlined in Section 2.4. Our goal in this section is to compare the out-of-sample performance, using the earnings calls from 2015–2020, of the different LM and ML dictionaries. We note that there is going to be some overlap between the ML and LM dictionaries, to the extent that the ML algorithm picks words that are part of the LM dictionaries. In order to compare the two dictionaries on equal footing, we will create three separate sentiment scores: one using LM words that do not overlap with the ML dictionaries, one using ML words that do not overlap with the LM dictionaries, and a last one that consists of the words that overlap across the LM and ML dictionaries. We give more details on coverage

[25] The results are not sensitive to these choices. The higher the threshold the smaller the potential overfit from the ML algorithm, at the cost of fewer signals. But the empirical results, both in terms of overall fit and economic significance, are very stable for other thresholds.

and size of these different dictionaries in Section 4.1.

In the first column in Table 2, we report the results using LM unigrams that do not overlap with the unigrams generated by the ML algorithm.[26] We find that both the positive and negative LM dictionaries significantly predict the stock market reactions. The statistical significance is strong, and the economic magnitudes are large: a one-standard deviation change to the positive (negative) sentiment score translates into a 0.41% increase (-0.50% decrease) in the stock price reaction. The R^2 of the regression increases from 1.7%, in a specification without any of the textual variables, up to 2.1% when including the two textual LM measures. We highlight how the LM dictionaries do fairly well in the earnings calls corpus, both the negative and the positive word lists, despite the original Loughran and McDonald (2011) paper developed them in the context of 10-K statements.

The second column in Table 2 repeats the exercise for the ML unigrams that do not overlap with the LM dictionaries. We see that the predictability is significantly stronger, with (absolute) t-stats above 7, and an adjusted R^2 at 4.6%, twice as high as with the LM words. The marginal effects are also stronger: a one standard deviation change in the positive (negative) sentiment scores results in increases (decreases) in the stock price reaction amounting to 0.98% (-1.37%), roughly 2.5 times bigger reactions that in column one.

The third column in Table 2 reports the estimates using the unigrams that overlap between the LM and ML dictionaries. The overall fit of the regression goes up, with (absolute) t-stats over 9, and an R^2 of 5.4%. The economic magnitudes of the coefficients are also larger: a one standard deviation change in the positive (negative) sentiment scores results in increases (decreases) in the stock price reaction amounting to 1.25% (-1.56%). While the LM words in general have some predictive power (column one), those that are also chosen by the ML algorithm are significantly stronger.

The fourth column in Table 2 considers ML bigrams. We see that the economic magnitude of the coefficients are quite large: a one standard deviation change in the positive (negative) sentiment scores result in increases (decreases) in the stock price reaction of 1.38% (-1.36%). The statistical significance is also stronger, with (absolute) t-stats over 8, and an overall R^2 of 4.5%. All these metrics are higher than using the LM words, with only the LM/ML overlap unigrams having roughly equal economic/statistical significance.

While the above univariate regressions are quite persuasive regarding the performance of the ML algorithm relative to the LM dictionaries, one cannot make further conclusions without a multi-variate analysis, included in columns five and six in Table 2. In column five we restrict

[26] In Table 10 in the Appendix we include the results including all control variables.

attention to ML unigrams, whereas in column six we study ML bigrams, controlling in both cases for the sentiment scores from the LM dictionaries. Focusing on column five, we find that the LM dictionaries load with the wrong sign when estimated jointly with the other sentiment scores: the LM positive has a negative coefficient, and LM negative has a positive (and statistically significant) coefficient.[27] On the other hand, both the ML positive and negative word lists have economically large coefficients (0.78 and -0.94) and associated t-stats (8.7 and -9.8). These are comparable to the overlap scores (LM & ML), which have slightly higher coefficients (0.90 and -1.32), and similar statistical significance (t-stats around 9).

The evidence from column five suggests that the ML dictionaries perform significantly better, with only the LM words that are validated by the ML algorithm having explanatory power. Furthermore, the LM words that do not overlap with ML seem to be associated with stock returns with the wrong sign, once we control for sentiment using the ML dictionaries. We also note that the joint regression in column five yields an R^2 of 6.5%, higher than all previous individual regression specifications.

The last column in Table 2 looks at the performance of the bigram dictionaries, estimated jointly with the LM word lists. We again find that the LM terms without overlap do not load in a significant way, with the LM negative again carrying the wrong sign. The joint LM & ML dictionaries have similar economic and statistical significance to previous specifications, both highly statistically significant. The bigram sentiment scores have similar performance as well, with the positive (negative) bigrams moving stock prices by 1.06% (-0.79%).

To summarize, in this section we have conducted an empirical exercise that starts with an arbitrary document-term-matrix (in a given n-gram space). We show how training using the early half of our sample, 2006–2014, allows us to construct strong predictors of price movements during our out-of-sample period 2015–2020. Our results show how the robust MNIR algorithm generates sentiment dictionaries that have much stronger contemporaneous correlations with stock returns, relative to the LM dictionaries. We turn next to see how each dictionary performs using other corpora, namely 10-K statements and WSJ articles.

3.2 External validity

Loughran and McDonald (2011) focus their dictionary construction using the corpus of 10-K statements, the annual reports filed by publicly traded firms in the EDGAR system. In contrast, our analysis has focused on the corpus from earnings calls. In this section we study to what extent the ML dictionaries generated using the robust MNIR algorithm can compete with the

[27] We note that this is driven by the inclusion of the LM & ML positive/negative scores, not by the ML scores, i.e. we find the same results when omitting the ML positive/negative variables.

LM dictionaries using other corpora, in particular the 10-K statement releases, as well as WSJ articles, the two databases discussed in Section 2.1.

We start by comparing the dictionaries generated by the ML algorithm developed in Section 2, fitted using the earnings calls corpus on the full sample (2006–2020), to the standard LM dictionaries. We choose the *n*-grams using the robust MNIR steps from Section 2.2, setting the criteria for inclusion as in Section 3.1. We then build sentiment scores using the new corpora (10-K statements or WSJ articles). Our empirical approach mimics the one from Section 3.1, we simply compare the external validity of the LM and ML dictionaries when applied to a different corpus. We note that while these two new corpora are quite related to the earnings calls corpus that we use to develop the ML dictionaries, the language used in verbal discourse (earnings calls) is certainly different than that used for regulatory filings (10-K) and journalists prose (WSJ). We also note that the stock price reaction to the 10-K releases and WSJ article publication is much more muted than the one to earnings calls (see Figure 1).

When using the full sample, the LM dictionaries and the ML dictionaries overlap on 18 (30) positive (negative) words. We emphasize how small these word lists are relative to the standard sentiment dictionaries. There are another 329 (2,315) positive (negative) LM words that do not overlap with the words picked by the ML algorithm. The unique positive (negative) unigrams stemming from the ML algorithm add up to 57 (64) tokens, also a significantly smaller number than the LM dictionaries. The ML positive (negative) bigrams amount to 381 (344) different terms.

In Table 3, we present the results using different dictionaries on the 10-K corpus, essentially replicating Table 2 with the 10-K corpus. The first column shows the (unique) LM sentiment scores are barely associated with the stock price reactions during the release of 10-K statements.[28] The (unique) LM positive dictionary score loads with a negative sign, and the (unique) LM negative dictionary score has a small economic impact, which is marginally statistically significant (a one standard deviation change in sentiment moves returns by 6 basis points, with a t-stat of -1.9). The (unique) ML positive unigrams have a 11 basis points impact on returns (t-stat 5.1), whereas we cannot reject the null that the ML negative unigrams are not related to returns. The joint LM & ML dictionaries, on the other hand, both load with the right signs, with absolute t-stats above 2. The ML bigram dictionaries have the strongest economic and statistical impact, as shown in column four: a one standard deviation change in the positive (negative) sentiment score is associated with a 15 (-16) basis points return change (t-stats of 3.8 and -2.6

[28] This is sample period specific. We can reproduce the results in LM to three significant figures using the sample period in their paper. The stock price reaction to 10-K releases has dropped significantly over the last decade, perhaps because earnings calls have gained in dissemination and visibility. See Li and Ramesh (2009) for similar estimates.

respectively).

When estimating jointly all the unigram dictionaries, presented in column five of Table 3, we see a similar pattern to the one in the univariate analysis. The (unique) LM negative dictionary loses all its predictive power, and the (unique) LM positive still loads with the wrong sign. On the other hand, the (unique) ML positive words have a 13 basis points impact (per one standard deviation change), with a t-stat of 4.9. The overlap dictionaries point estimates, and statistical significance, are similar to those estimated in the univariate specification, having impact of 5 basis points on the positive domain, and -21 basis points on the negative domain. Finally, column six confirms the results with bigrams are the strongest, very similar to those in the univariate specifications.

One could have conjectured that the reason the ML dictionaries outperform the LM dictionaries in our analysis in Section 3.1 is due to the fact that the LM were developed for 10-K statements, not for earnings calls. On the other hand, we have shown that the LM dictionaries predictability for earnings calls is quite strong (see Table 2), and that their predictability on 10-K statements is rather weak, relative to the ML word lists. While the 10-K statement releases are not a particularly powerful event study, we still find that the ML dictionaries capture more color than the words in the LM dictionaries. Even when working with the corpus from 10-K statements, the origin of the LM word lists, the evidence in Table 3 shows that the ML algorithm can capture some sentiment that is not measured by existing bag-of-word approaches.

In Table 4 we repeat our main empirical exercise using WSJ articles as the event of interest. The first four columns present the univariate estimates, as in Tables 2-3. The LM (unique) dictionaries both load with similar economic magnitudes: 10 (-11) basis points impact (per standard deviation change in sentiment) for the positive (negative) sentiment scores, with t-stats above 7. We note the R^2 is quite low in these specifications, at 0.7%. The (unique) ML dictionaries, and the overlap dictionaries (LM & ML) perform much better, with R^2 of 1% and 0.9% respectively, and economic magnitudes of 12 and 16 basis points in the positive domain, and -21 and -19 in the negative domain. The ML bigrams also have larger point estimates than the LM words, with marginal impacts of 14 basis points per standard deviation shock, both in the positive and negative domain. While the LM dictionaries perform better on the WSJ corpus than with the earnings calls and 10-K corpora, the ML word lists have a clear edge on these univariate specifications on the WSJ corpora, with higher t-stats, economic magnitudes, and overall statistical fit (R^2), corroborating their external validity.

In the joint estimations presented in columns five and six of Table 4, the univariate results seem to be quite robust. When looking only at unigrams (column five), we find that all three unigram lists (LM only, ML only, joint LM/ML) have some marginal explanatory power, with larger economic and statistical magnitudes for the ML sentiment scores. On the positive side, we see the effects are 8, 9 and 12 basis points (for LM, ML, LM&ML), whereas on the negative side we have point estimates of -5, -17, and -16 basis points. The joint specification shown in column six, which considers ML bigrams and LM unigrams, as in previous tables, replicates the univariate results, with all three dictionaries having some marginal significance, with similar magnitudes as in columns 1–4. It is worthwhile noticing how the joint fits in columns 5–6 almost double the R^2 of the regression with only LM words, even while the events are not as impactful as when reading the earnings calls corpus (Table 2). While the LM dictionaries do seem to have some marginal explanatory power in the WSJ corpus, the ML dictionaries, both using unigrams and bigrams, bring further color to journalists' prose, with economic magnitudes 2–3 times bigger, and higher statistical significance.

To summarize, the evidence in this section shows how the ML dictionaries constructed with the earnings calls corpus following the robust MNIR model discussed in Section 2.2 have strong external validity. Both the unigram dictionaries, which contain a handful of words relative to the existing bag of words, and the bigram dictionaries have stronger contemporaneous correlations with stock returns. In the rest of the paper we attempt to further compare the human dictionaries (LM) versus those from the ML algorithm, and we will try to understand what drives the performance improvements.

4 Coloring words

The results in Section 3 suggest that large document-term-matrices trained using stock price reactions are great predictors out-of-sample. We dig into what drives our improvements in predictability in this section. In Section 4.1 we look at the breadth of our dictionaries, changing the size of the dtms we use. Section 4.2 compares the LM and ML dictionaries in detail. In Section 4.3 we study the disambiguation of unigrams that the ML algorithm creates when working with bigrams, studying the role of negation in Section 4.4. In Section 4.5 we discuss the data depository we provide to complement our paper, in particular the different dictionaries a reader may want to implement with our output.

4.1 Dictionary breadth

Our starting point is a summary of the text for each earnings call as a document-term-matrix with a given set of tokens. In the analysis in Section 3 we use the top 16K (65K) unigrams (bigrams) by frequency. A natural question to ask is to what extent these representations cover the whole corpus, and what are the frequencies of *n*-grams that we are studying, relative to those in the standard bag-of-words approach. We add the statistics on trigrams for completeness, and to document some of the reasons bigrams seem to be as high-order as one should go for sentiment analysis purposes.

The top panel of Figure 2 plots the percentage of the corpus that is covered by dtms with 2^k terms, for $k = 9, \ldots, 26$. The red crosses correspond to the unigram representation: we see that with as few as 4–8K unigrams we are reading virtually the entirety of the earnings calls corpus. This is in contrast with the bigram coverage (in blue): even with 10K tokens the dtm only covers about 28% of the corpus. One has to use dtms with more than 65K tokens to cover about 50% of the corpus with bigrams. For trigrams, in green, the coverage with 10K terms is below 10% of the corpus, and one needs to have dtms with over 500K tokens in order to cover more than 25% of the corpus.

The bottom panel of Figure 2 plots the rank-frequency distribution for uni/bi/trigrams. We note how the unigram and bigram lines cross around the 4,000 mark, i.e. the 4,000th bigram by frequency shows up in the earnings calls more frequently than the 4,000th unigram. For trigrams that crossing point is around the 10,000th ranked token. Most importantly, while unigrams do fall down significantly after the first few thousand words, bigrams and trigrams have significantly thicker tails: the 50,000th bigram (by frequency) still has several hundred appearances in the earnings calls corpus. With a set of events in the 60K range, the sentiment of such *n*-grams is not easy to estimate, but the ML algorithm attempts to color them.

The average number of unigrams per call is 3,145, with an average of 1,030 unique words. The average number of bigrams is 2,785, with an average of 2,430 unique bigrams. For trigrams the numbers are very similar to bigrams: the total number of trigrams per call are 2,455, whereas the average number of unique trigrams is 2,376.[29] There are about 2.4 times as many unique bigrams than unigrams in a given call, but the number of unique trigrams is very similar to that of bigrams at the earnings call level. This is true despite the fact that there are significantly more unique trigrams (78m) than unique bigrams (15m) across the whole corpus.[30]

At the earnings call level, it seems like the document is well summarized using bigrams, without needing to use trigrams. At the same time, the full bigram representation is significantly larger than that of unigrams, by a factor of almost 100, despite all the cleaning/token removal we

[29] Since we tokenize at the sentence level, there are fewer trigrams than bigrams, and fewer bigrams than unigrams (for every n word sentence, we have $n-1$ bigrams and $n-2$ trigrams).

[30] The number of unique unigrams is 186,994. It is worthwhile noticing these numbers are “large,” as standard estimates of English native speakers dictionaries are around 20,000 words. This is mostly due to the nature of the corpus, composed of transcriptions of the earnings calls which are going to have typos and often very specialized language.

perform. The fact that the number of trigrams (per call) is similar to bigrams also hints at staying only with bigram dtms: the potential marginal gains do not outweigh their lack of coverage show in the top Panel of Figure 2.[31] This suggests our analysis is quite comprehensive, within the standard bag-of-words approach.

The above discussion gives some important aspects of the basic construction of our algorithm, which relies on starting with a dtm. But it does not speak to the final breadth of the dictionaries that come out of the robust MNIR algorithm. As in the analysis in Section 3, we group the unigrams as LM (only), ML (only), and the joint LM/ML lists. In Table 5 we present such measures, reporting the total number of *n*-grams in each dictionary, as well as the percentage of a corpus that is covered by a given dictionary. We consider the three corpora analyzed in Section 3, where the ML dictionaries are constructed using the full earnings calls corpus (2006–2020).

In the first Panel of Table 5 we look at the positive word lists. We see that the 329 LM positive words amount 1.9% of the earnings calls corpus, 0.7% of 10-Ks, and 1.3% in the WSJ. On the other hand, we see that the 57 ML positive words cover roughly 3–4 times more: 8.4% in the earnings calls, 4.1% in 10-Ks, and 3.2% in the WSJ. When looking at the list that overlaps, a total of just 18 words, we see that they cover a non-trivial amount of the corpora, comprising 1.3% of the earnings calls, 0.2% of 10-Ks, and 0.4% in the WSJ. Finally, we see that the ML positive bigrams are well represented in the earnings calls corpus (2.3%), with slightly lower coverage in 10-Ks/WSJ (0.8% and 0.3%), as expected.

The coverage numbers for negative dictionaries, presented in the bottom panel of Table 5, paint a similar picture. The 2,315 LM negative words cover slightly more of the 10-K and WSJ corpus (2.7% and 3.2%), relative to the earnings calls (1.4%), and also relative to the LM positive words. The 64 ML negative words are nonetheless more frequent on both earnings calls and 10-K statements (4.5% and 4.3%), and they have a similar coverage to the LM words in the WSJ corpus (3.1%). The list of negative overlap words, 30 different unigrams, covers about 0.4–0.5% of the three corpora, a non-trivial number given the (small) size of the dictionary. The ML negative bigrams are slightly less prevalent than the positive bigrams, but still cover 1.4% of the earnings calls, and 0.7% (0.4%) of 10-Ks (WSJ).

The robust MNIR relies on repeated positive/negative signals across different validation samples. The 80% (45%) inclusion criteria for unigrams (bigrams) is quite stringent, picking a small set of tokens that have consistency across different subsamples (industry/time). On the unigram side, our results in Section 3 are driven by 75 positive words, and 94 negative words,

[31] A previous draft of the paper confirmed this: the fits using trigrams are significantly worse than those using bigrams.

significantly less (in terms of counts) than those in standard sentiment dictionaries. The 0.80 cutoff we use for the inclusion of unigrams corresponds roughly to the 99.5% percentile of the D^+ and D^- scores, using a dtm of size 16K. The 0.45 cutoff we use for bigrams similarly corresponds (roughly) to the 99.5% percentile of the D^+ and D^- scores, using a dtm of size 65K. The evidence in Table 5 shows that not only are the ML words strong signals, they are also much more common than the LM dictionaries, despite the fact that the unigram lists are significantly smaller in size. Even the bigrams lists are fairly small in size, around 350 tokens for both positive and negative, which generate about the same number of signals as the LM dictionaries in the earnings calls corpus. We do note that the ML dictionaries, across the different groups, always exhibit a larger coverage in the earnings calls corpus relative to either 10-Ks or WSJ articles: they were constructed using earnings calls, thus coverage due to context specificity will be higher.

4.2 Comparing the LM and ML dictionaries

Our next exercise is to study more carefully the actual choices of positive and negative labels coming from the machine learning algorithm, and how they compare to the LM dictionaries.

In Table 6 we present the top 30 positive and negative words in the LM dictionaries by frequency, together with their associated robust MNIR scores. We note that these 60 LM words cover more than 65% of the total term frequencies of all LM words in the earnings calls corpus.[32] The table lists the token in consideration, its coverage over the whole corpus (Cov., measured in basis points), and the percentage of the 500 cross-validation samples for which the unigram is labelled as positive (negative).

The ML algorithm broadly agrees with the LM classification. Of the 30 positive LM words listed in the table, 12 are also classified as positive using the robust MNIR method: the top three LM words by frequency are a leading example (*good*, *strong*, *better*). It is interesting to see that most of the words associated with the verb *improve* get classified as positive both by the LM and ML algorithms, with the exception of the infinite form *improve* itself: *improvement*, *improved*, *improving* and *improvements* are positive according to the robust MNIR algorithm, but *improve* is negative in 34% of the cross-validation samples (positive only in 11% of them). Several other LM unigrams have relatively high ML scores (*opportunity*, *progress*), but many others are not that positive at all (*best*, *despite*), and the term *confident* is actually included in

[32] This is not driven by differences in the earnings calls corpus and 10-K statements. Using 10-K statements, we find that the top 50 LM positive words cover 80% of all the positive term frequencies, and the top 200 LM negative words cover more than 80% of all the negative term frequencies.

the ML negative dictionary.[33] The ML and the LM dictionaries are labelling positive words quite differently.

Similar agreements between LM and ML can be found in the negative domain for words such as *decline(d)*, *loss* or *challenges(ing)*. There is some disagreement due to external validity, i.e. *question(s)* is a very special word in earnings calls. But there is plenty of disagreement: *break* scores positively by the ML algorithm, and *restructuring* is a toss-up (31% positive and 30% negative ML scores). Other words such as *recall*, *against*, and *volatility* are not particularly negative according to the ML scores. Our approach captures color of finance discourse that is not measured by the standard bag-of-words approaches.

Table 7 considers the ML words, ranked by frequency, essentially mimicking Table 6 but focusing on the set of words chosen by the ML algorithm. The first thing to note is that the frequency counts of the ML words are significantly higher, echoing the evidence from Table 5. There is not that much overlap of LM words in Table 7, relative to Table 6, as of the top 30 ML positive words only five (*good*, *strong*, *better*, *improvement* and *pleased*) are in the LM dictionaries, with only three of the top ML 30 negative (*decline*, *loss*, and *negative*) being part of the LM dictionaries.

The threshold of 80% that we use to classify words as positive (negative) is fairly stringent, it includes only 75 positive and 94 negative words (Table 5). And the bulk of the signals, in terms of term frequencies, are included in Table 7. While a discussion of each term is beyond the scope of our paper, we highlight a few of the choices made by the robust MNIR algorithm.

The top positive word is *think*, not a term likely to be included by a human as particularly positive or negative. Within the top five negative words, we find *believe*, which is semantically quite related. Our research argues that these two relatively common words are being used in different contexts. The bigrams *believe important*, *continue believe*, *still believe* all have high term frequency counts, and are flagged as negative by the ML algorithm. On the other hand, *think continue*, *think kind*, *think really* are both frequent and quite positive, according to the ML algorithm.

On a similar spirit, we see the token *increase(d)* is labelled as positive by the ML algorithm, while *decrease* is considered negative. While these seem like natural choices, ex-post, neither is included in the LM dictionaries. Context is again important: the use of the verb increase is associated with positive changes, while the verb decrease is hard to relate to any positive event.

[33] We note that the updated 2020 LM list excludes 17 terms, relative to the original version. Among these 17 terms we have *great* and *benefit*, both of which are part of the ML positive dictionaries, as shown in Table 7.

There are some natural words that seem positive in Table 7, even if a human may hesitate to consider unambiguously positive:[34] *growth*, *up*, *well*, *over*, *really*, *continue(d)*, *increase(d)*, *lot*, *great*, *across*. There are a fair number of accounting/finance related words in Table 7, *share*, *cash*, *operating*, *margin*, *income*, *flow*, hinting at the fact the managers may want to discuss such facts when things are going well.

On the negative side, we see the top negative word (by frequency) is *not*, which is a strange choice by the ML algorithm, but with an unambiguously negative ML score (in 97% of the cross-validation samples it is consider negative by the MNIR algorithm). We will further analyze negation in Section 4.4.

There are several other very frequent words that do have a strong negative sentiment, but are not included in the LM dictionaries: *down*, *back* and *impact(ed)* are leading examples, all associated with bigrams that are quite negative, as we will illustrate in Section 4.3. Using the words *change(s)*, *costs*, *timing* also has a negative sentiment, as do references to *expectations*, *expected*, *trying*, and *understand*.

The main goal of the discussion in this section, associated with Tables 6 and 7, is to show how while there is some agreement between the LM and ML dictionaries, they are coloring words in quite different ways. Their intersection is particularly powerful, as shown by the LM & ML scores throughout Tables 2-4, while it is a small set of words (Table 5). But the ML words not included in the LM lists are both much more frequent, and also quite colorful. In the next section we will use bigrams to illustrate how the ML algorithm makes such word choices.

4.3 Disambiguation

The goal of this section is to study the role of bigrams to construct measures of sentiment above and beyond the standard “bag-of-words,” which focuses on unigrams, as well as highlight why the ML algorithm chooses the words it chooses. We argue in this section that using the color of bigrams helps understand the sentiment of individual unigrams, and that bigrams are extremely useful at disambiguating the meaning of words.

We start our analysis by depicting graphically the main output from the ML analysis: the percentage of times a given unigram is considered positive/negative/neutral. Figure 3 plots these three coordinates, for the top 4,096 unigrams by frequency, in a ternary plot. We plot the positive (negative) LM terms in blue (red), and the rest in black.

Under the null hypothesis that there is no need for disambiguation, we would expect all points

[34] One can *increase* costs, and *continue* to do poor things, but no CEO would use such narratives in an earnings call.

to concentrate in the three corners: either a words is neutral, positive, or negative. Figure 3 shows that the ML algorithm strongly rejects this null: the bulk of the points is concentrated in the upper center of the triangle, corresponding to unigrams that are mostly neutral, with some instances of positive and negative context.

There are terms that have some zero positive (negative) scores across all of the samples, plotted on the sides of the triangle, but we see that the ML algorithm classifies many such terms as neutral in the majority of the cases. It is important to note that these patterns apply broadly to the LM dictionary words: on average the blue (red) words are on the positive (negative) side of the ternary plot, but by and large they concentrate on the top neutral space.

There are terms that seem to be fairly unambiguous, those in the shaded areas in the ternary plot, where the D^{+} and D^{-} scores are higher than our 0.80 threshold. Those are the terms that the ML algorithm picks, dropping all the other unigrams, which are mostly neutral. It is important to note that while there are a handful of LM terms in the shaded areas of Figure 3, most of these unigrams are new to the literature, and present one of the contributions of our paper.

As shown in Figure 3, there are many unigrams which have mixed sentiment loadings. The ML algorithm is able to disambiguate many of these using bigrams, coding differently a unigram according to its company. Table 8 considers bigrams associated with six unigrams, two from the LM dictionaries, and four from the ML dictionaries. We require the bigrams D^{+} (D^{-}) scores to be above 20%, and cap the list at five bigrams, since those will tend to dominate the sentiment scoring using frequency counts. The table lists the bigram, its relative frequency (out of all occurrences of a given unigram in our bigram dtm, the percentage of times that given bigram occurs), as well as its D^{+} score in the left panel, with the D^{-} scores in the right.

As argued earlier, the LM positive term *improve* does not score very positive in the ML analysis. We see that the bigrams *continue(s) improve* and *able improve* get relatively high ML positive scores. On the other hand, there are many bigrams that get flagged as negative by the robust MNIR estimation: *improve performance/over/second*, *going/conditions improve*. The existence of such sentences, in which *improve* is not being used with a positive connotation, makes the unigram not well suited as a sentiment indicator. Using bigrams, we get to pick only those instances where *improve* is indeed being used with a positive spin.

Next in the table we see the bigrams associated with *confident*. While there are two bigrams that are relatively positive, the most salient, is the bigram *remain confident*, which is very frequent (about 16% of all confident occurrences), and has a very large D^{-} score.

The words *solid* and *soft* both belong to the final ML dictionaries. Table 8 shows that these words are virtually associated with bigrams that are only positive (in the case of *solid*) or negative (in the case of *soft*). Note that the first *solid* bigrams are more than 20% of the frequencies of that term, whereas there is only one (rare) bigram with a D^- score above 20%. The three *soft* bigrams on the right panel comprise more than 30% of the counts of the term, all with large negative scores. The fact that these words are used in an unambiguously positive/negative sense makes them ideal candidates for sentiment dictionaries.

When looking at the positive bigrams associated with *cash*, we see how the five included comprise over 40% of its term frequencies. On the negative side, we see terms that most humans would associate with a negative connotation, i.e. *cash burn*, but their term frequencies are significantly lower, with none of the bigrams comprising more than 1% of all instances of cash.

The last term included in Table 8 is *continue*, which would hardly be considered as a sentiment word by most human readers. The bigrams shown indeed read quite "plain," but the ML positive scores, as measured by D^+, are quite large, with a couple in the 99.5% tail of D^+ scores.[35] Moreover, the term frequencies of those bigrams add up to over 15%, whereas those in the negative bigram list do not break 2%. The ML algorithm picks up *continue*, and derivatives of the verb, as a positive word precisely because its frequent use in a negative context.

The above discussion is limited, due to the large scope of words we study and obvious space constraints. Our goal is that it gives a feel for how and why the ML algorithm chooses terms to be included in positive/negative dictionaries. The data and code we share in the data depository discussed in Section 4.5 allows the reader to refine and expand the analysis in this section to look at any other word that is included in our analysis.

We end this section by highlighting some of the tradeoffs and limitations of our approach. Our ML algorithm is trained on earnings calls, which induces some context specificity (i.e. *cash flow*), which is overall positive, as we are trying to measure "finance discourse," but clearly not ideal: while a CFO will use *cash flow* during an earnings call only when things as going well, a journalist may choose to use it in a different context. We have calibrated our empirical exercise to avoid overfitting, but it is inevitable to have some words that may not resonate to a human. On the other hand, relaxing our stringent inclusion criteria will add many potentially good signals, at the cost of including other (noisy) terms.

[35] The bigrams "think continue" and "continue improve" do not make the table, but both have a relative frequency of 1.2%, and D^- scores of 73 and 64 respectively.

4.4 Negation

The token *not* deserves some further discussion, as it is standard negation in English, it is very common in our corpus, and the robust MNIR algorithm includes it in the ML negative word lists due to its D^{-} score of 0.97 (see Table 7), a clearly negative sentiment score. The folklore in the literature is that positive words have less impact due to such negations. Loughran and McDonald (2020), summarizing the literature, write: "The framing of negative information is so frequently padded with positive words that the measured positive sentiment is ambiguous. Although some papers have identified statistically significant effects associated with positive tone (e.g., García (2013), Jegadeesh and Wu (2013)), Tetlock (2007) and Loughran and McDonald (2011) find little incremental information in positive word lists, which is consistent with the concern about negation of positive words."

Table 9 mimics the construction of Table 8, focusing on bigrams that start with "not." The common wisdom echoed in the previous quote from Loughran and McDonald (2020) stems from the possibility, in the English language, to use negation to invert the meaning of a term. Focusing first on the right panel of Table 9, we see that there are four unigrams that are preceded by *not* that are ML negative bigrams, with fairly large D^{-} scores (*not able/happy/satisfied/pleased*). At the same time, there are two LM negative words that when negated are still being scored as negative by the ML algorithm (*not lost/losing*). Moreover, we have several ML negative unigrams that are preceded by not in Table 9 (*changed*, *believe*, *expected*, *issue*, *related*, *offset*), and their ML scores are still very negative.

Turning to the left panel, we see a similar mix. There is one single LM negative term that shows up as a positive bigram, *not break*. There are five positive terms that negated are still considered positive by the ML algorithm. Most importantly, the frequency counts of the bigrams in the left panel are significantly smaller than the frequency counts in the right panel. Furthermore, note how the D^{-} scores are quite large, relative to the D^{+} scores. The *not* bigrams are overall quite negative.

It is true that negating positive words generates negative bigrams, but it is also true that negating negative words generates negative bigrams. Differently put, using negation in English carries a strong negative sentiment, no matter what is being negated. At the very least this is what stock prices, read using our ML algorithm, suggest about negation in a financial context.[36]

[36] We are speculating in this footnote, but while we clearly agree much of the jargon we provide in our new ML dictionaries is context specific (finance/accounting/economics), the negation evidence we provide is likely to have more external validity. For example, the best-selling book "How to negotiate with your kids" is explicit about how using *not* while talking to children is perceived quite negatively.

4.5 Data depository

The data depository[37] that complements the paper consists of the underlying dtm representation of the earnings calls under study, with associated public metadata, together with the above dictionaries and other auxiliary files (code+). We note that we provide a version of our analysis that uses Kaggle data, which can be used to both train/predict (without some controls).[38] We include in our depository the code that generates the dictionaries introduced above, so the readers can adapt it to their needs. We also include functions that can reproduce the disambiguation results, as in Table 8.

Perhaps of most interest for readers wanting to go beyond the word lists included in the Appendix, we provide two files containing the robust MNIR output, the D^+ and D^- scores, for all unigrams in our 16K dtm, as well as all the bigrams in our 65K dtm. These files allow researchers to be more stringent/lax regarding the inclusion of unigrams or bigrams in their own projects, as well as reproduce many of the results we report in our paper.

In case our English narrative in the paper is not persuasive enough, we hope the open source code and data we provide can convince the interested reader that, while ML algorithm does not speak English, it brings out tons of color to financial discourse. We conjecture, but leave for future research, that the approach advocated in our research should work equally well in other languages/emojis.[39]

5 Conclusion

We construct dictionaries based on a variation of the machine learning algorithm of Taddy (2013), using a large corpus of earnings calls transcripts. We find that the tokens chosen by our algorithm perform significantly better than the existing techniques based on bag-of-words. We further argue that the machine learning approach can help us refine existing word lists, highlighting which words have more bite than others, and also find new words that could be missed by human coders. Our empirical results show how bigrams can color financial text via disambiguation. While the debate is far from settled, our evidence shines a much brighter light on machine learning algorithms than that suggested in Loughran and McDonald (2020).

[37] See http://leeds-faculty.colorado.edu/garcia/data.html.

[38] We can reproduce all our results with this alternative dataset/empirical approach. See https://www.kaggle.com/borismarjanovic/price-volume-data-for-all-us-stocks-etfs.

[39] We note that working with sentiment dictionaries in other languages is a challenge, as translating the LM words is not a real option, given the nuances of translation. The robust MNIR algorithm is an off-the-shelf alternative to construct such sentiment dictionaries in any language, given a training sample associated with stock returns.

References

Antweiler, W., and M. Z. Frank, 2004, "Is all that talk just noise? The information content of internet stock message boards," *Journal of Finance*, 59(3), 1259–1294.

Baker, S. R., N. Bloom, and S. J. Davis, 2016, "Measuring economic policy uncertainty," *Quarterly Journal of Economics*, 131, 1593–1636.

Bochkay, K., R. Chychyla, and D. Nanda, 2019, "Dynamics of CEO disclosure style," *Accounting Review*, 94(4), 103–140.

Bybee, L., B. T. Kelly, A. Manela, and D. Xiu, 2019, "The structure of economic news," working paper, Yale University.

Chen, J. V., V. Nagar, and J. Schoenfeld, 2018, "Manager-analyst conversations in earnings conference calls," *Review of Accounting Studies*, 23, 1315–1354.

Cong, L. W., T. Liang, and X. Zhang, 2020, "Textual factors: A scalable, interpretable, and data-driven approach to analyzing unstructured information," working paper, University of Chicago.

Cookson, J. A., and M. Niessner, 2020, "Why don't we agree? Evidence from a social network of investors," *Journal of Finance*, 75(1), 173–228.

Das, S. R., and M. Y. Chen, 2007, "Yahoo! for Amazon: Sentiment extraction from small talk on the web," *Management Science*, pp. 1375–1388.

Fama, E. F., and K. R. French, 2001, "Disappearing dividends: Changing firm characteristics or lower propensity to pay?," *Journal of Financial Economics*, 60(1), 3–43.

Fama, E. F., and J. MacBeth, 1973, "Risk, return, and equilibrium: Empirical tests," *Journal of Political Economy*, 81, 607–636.

Fedyk, A., 2020, "Front page news: The effect of news positioning on financial markets," working paper, University of California Berkeley.

———, 2021, "Disagreement after news: Gradual information diffusion or differences of opinion?," *Review of Asset Pricing Studies*, 11(3), 465–501.

Feldman, R., S. Govindaraj, J. Livnat, and B. Segal, 2010, "Management's tone change, post earnings announcement drift and accruals," *Review of Accounting Studies*, 15(4), 915–953.

Frankel, R., M. Johnson, and D. J. Skinner, 1999, "An empirical examination of conference calls as a voluntary disclosure medium," *Journal of Accounting Research*, 37(1), 133–150.

García, D., 2013, "Sentiment during recessions," *Journal of Finance*, 68(3), 1267–1300.

García, D., and Ø. Norli, 2012, "Geographic dispersion and stock returns," *Journal of Financial Economics*, 106(3), 547–565.

Gentzkow, M., B. Kelly, and M. Taddy, 2019, “Text as data,” *Journal of Economic Literature*, 57(3), 535–574.

Glasserman, P., and H. Mamaysky, 2019, “Does unusual news forecast market stress?,” *Journal of Financial and Quantitative Analysis*, pp. 1–38.

Goldman, E., N. Gupta, and R. Israelsen, 2022, “Political Polarization in Financial News,” working paper, Indiana University.

Griffin, P. A., 2003, “Got information? Investor response to form 10-K and form 10-Q EDGAR filings,” *Review of Accounting Studies*, 8, 433–460.

Hanley, K. W., and G. Hoberg, 2012, “Litigation risk, strategic disclosure and the underpricing of initial public offerings,” *Journal of Financial Economics*, 103, 235–254.

Hansen, S., M. McMahon, and A. Prat, 2018, “Transparency and deliberation within the FOMC: A computational linguistics approach,” *The Quarterly Journal of Economics*, 133(2), 801–870.

Hoberg, G., and C. Lewis, 2017, “Do fraudulent firms produce abnormal disclosure?,” *Journal of Corporate Finance*, 43, 58–85.

Hoberg, G., and G. Phillips, 2016, “Text-based network industries and endogenous product differentiation,” *Journal of Political Economy*, 124(5), 1423–1465.

Israel, R., B. Kelly, and T. Moskowitz, 2020, “Can machines “learn” finance?,” *Journal of Investment Management*, 18(2).

Jegadeesh, N., and D. Wu, 2013, “Word power: A new approach for content analysis,” *Journal of Financial Economics*, 110, 712–729.

Ke, Z. T., B. T. Kelly, and D. Xiu, 2019, “Predicting returns with text data,” working paper, University of Chicago.

Kelly, B., A. Manela, and A. Moreira, 2018, “Text selection,” working paper, Washington University at St Louis.

Kogan, S., D. Levin, B. R. Routledge, J. S. Sagi, and N. Smith, 2009, “Predicting risk from financial reports with regression,” *North American Association for Computational Linguistics Human Language Technologies Conference*.

Larcker, D. F., and A. A. Zakolyukina, 2012, “Detecting deceptive discussions in conference calls,” *Journal of Accounting Research*, 50(2), 495–540.

Li, E. X., and K. Ramesh, 2009, “Market reaction surrounding the filing of periodic SEC reports,” *The Accounting Review*, 84(4), 1171–1208.

Loughran, T., and B. McDonald, 2011, “When is a liability not a liability? Textual analysis, dictionaries, and 10-Ks,” *Journal of Finance*, 66, 35–65.

——— , 2016, “Textual analysis in accounting and finance: A survey,” *Journal of Accounting Research*, 54, 1187–1230.

——— , 2020, “Textual analysis in finance,” working paper, Working paper, University of Notre Dame.

Manela, A., and A. Moreira, 2017, “News implied volatility and disaster concerns,” *Journal of Financial Economics*, 123(1), 137–162.

Matsumoto, D., M. Pronk, and E. Roelofsen, 2011, “What makes conference calls useful? The information content of managers’ presentations and analysts’ discussion sessions,” *Accounting Review*, 86(4), 1383–1414.

Meursault, V., P. J. Liang, B. R. Routledge, and M. M. Scanlon, 2021, “PEAD.txt: Post-earnings-announcement drift using text,” working paper, Federal Reserve Bank of Philadelphia.

Muslu, V., S. Radhakrishnan, K. Subramanyam, and D. Lim, 2015, “Forward-looking MD&A disclosures and the information environment,” *Management Science*, 61(5), 931–948.

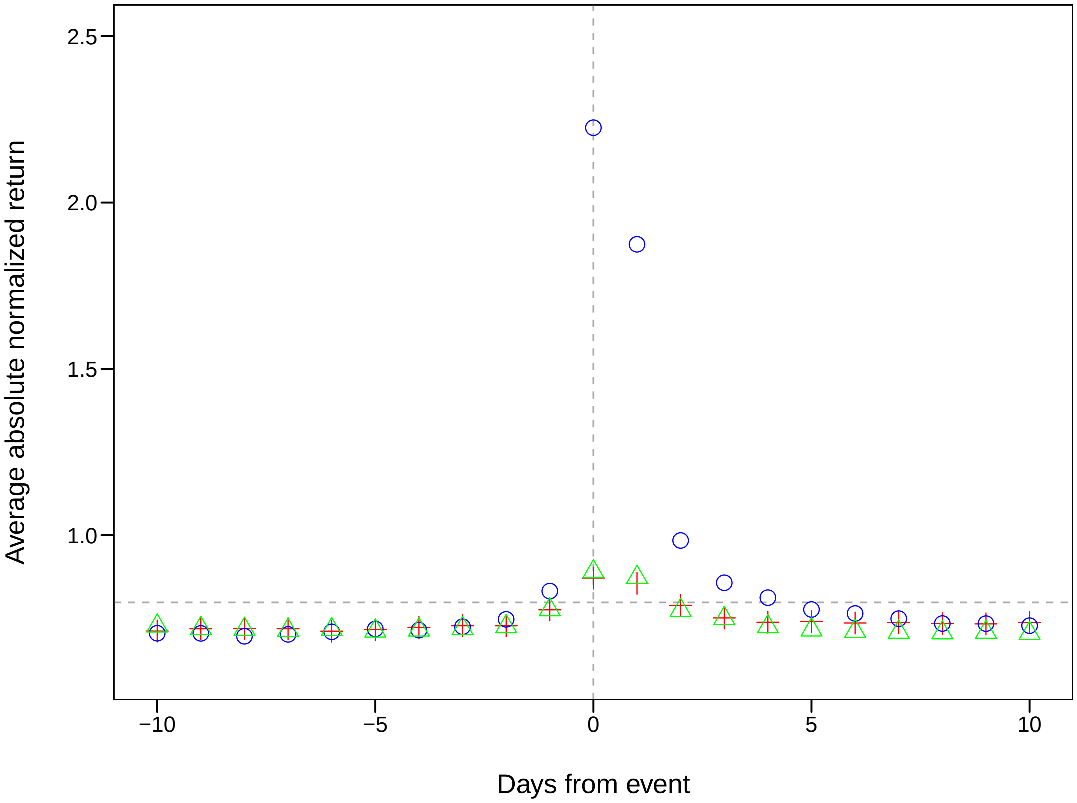
Rabinovich, M., and D. Blei, 2014, “The inverse regression topic model,” in *Proceedings of the 31st International Conference on Machine Learning*, ed. by E. P. Xing, and T. Jebara, vol. 32 of *Proceedings of Machine Learning Research*, pp. 199–207, Bejing, China. PMLR.

Roberts, M. E., B. M. Stewart, D. Tingley, E. M. Airoldi, et al., 2013, “The structural topic model and applied social science,” in *Advances in Neural Information Processing Systems Workshop on Topic Models: Computation, Application, and Evaluation*, vol. 4. Harrahs and Harveys, Lake Tahoe.

Solomon, D., 2012, “Selective publicity and stock prices,” *Journal of Finance*, 67(2), 599–637.

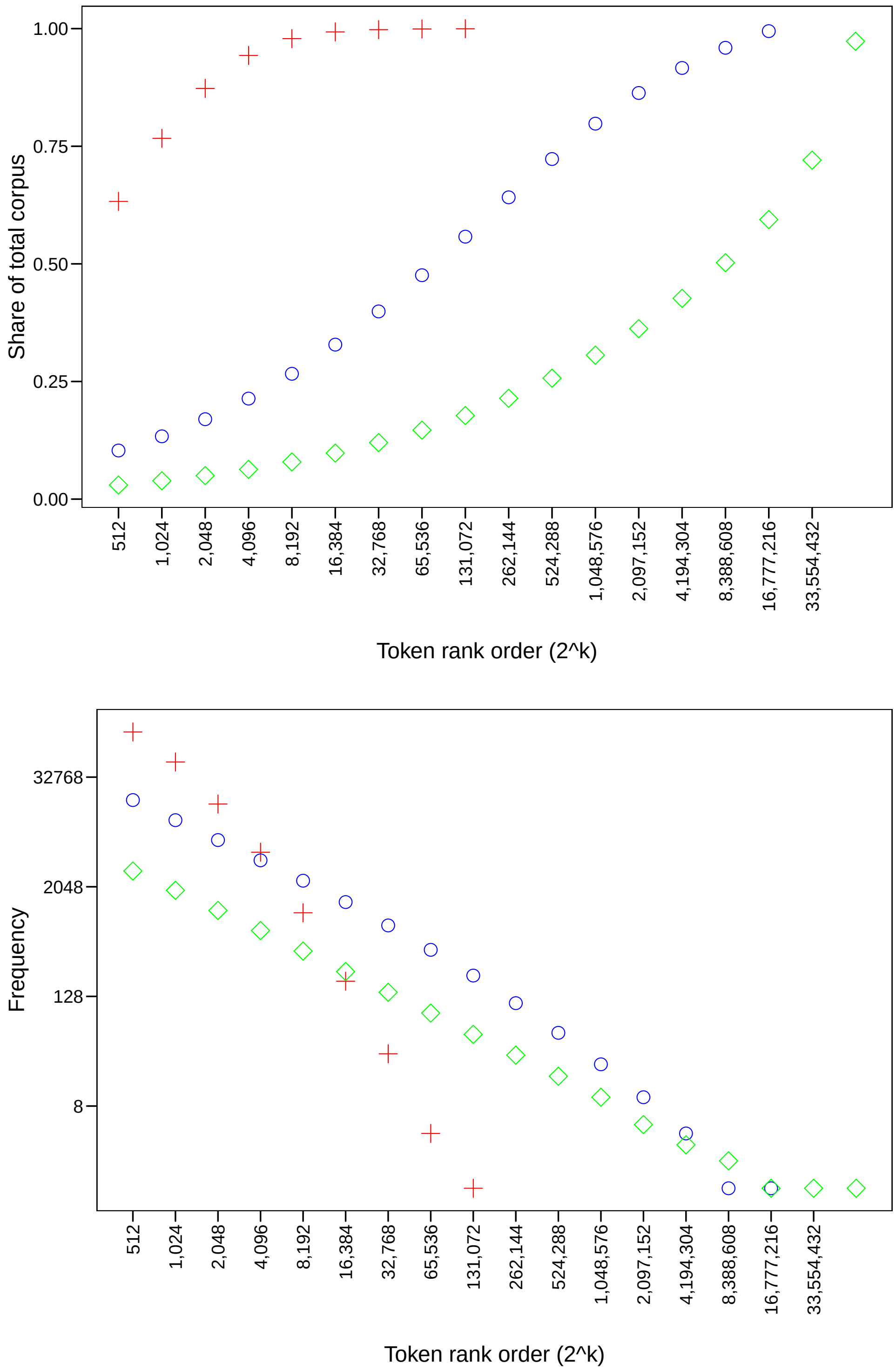
Taddy, M., 2013, “Multinomial inverse regression for text analysis,” *Journal of the American Statistical Association*, 108(503), 755–770.

Tetlock, P. C., 2007, “Giving content to investor sentiment: The role of media in the stock market,” *Journal of Finance*, 62(3), 1139–1168.



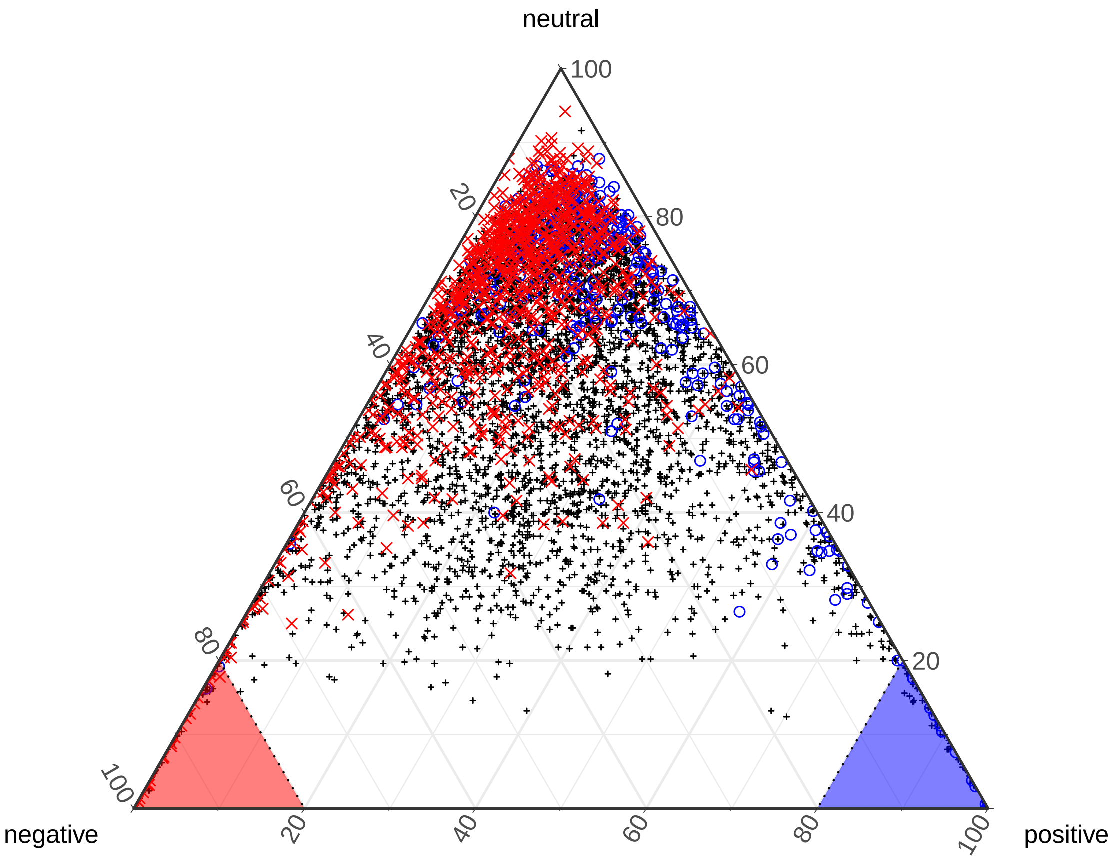
This figure reports average absolute normalized excess return around the earnings call events (blue), the filing date of 10-Ks (green) and WSJ article publications (red). Excess return is CRSP daily stock return less the value-weighted total return index, normalized by its mean and standard deviation, computed in the period of -60 to -2 days relative to the event date. The dash horizontal line is the the expectation of the absolute value of a standard normal random variable.

Figure 1: Average absolute returns around events (earnings calls, 10K, WSJ)



The top graph plots the proportion of the total text of earnings calls that is covered by having document-term-matrices of different sizes, starting with 512 tokens (2^9) up to 67m (2^{26}) tokens. The bottom graph plots the log-frequencies when ranking individual *n*-grams by such frequencies. The red crosses refer to unigrams, the blue circles to bigrams, and the green diamonds to trigrams.

Figure 2: *n*-gram coverage and log-frequencies



This ternary graph plots each of the top 4,096 unigrams by frequency, as well as all LM words, showing the percentage of times a given unigram is considered positive (right), negative (left), or neutral (top), according to the robust MNIR algorithm. The blue circles are LM positive words, whereas the red x's are LM negative words. The black crosses are terms that do not belong to the LM dictionaries.

Figure 3: Unigram and bigram sentiment scores

Table 1: Corpus overview

The following three panels show the start and end dates for each of the three corpora we study, as well as the number of unique firms and total number of observations (event-firm).

Earnings calls	
Start	13.10.2005
End	07.10.2020
Unique firms	3,229
Observations	85,530

Annual reports (10-K)	
Start	02.01.1996
End	27.12.2018
Unique firms	10,076
Observations	76,922

Wall Street Journal (WSJ)	
Start	03.01.2000
End	31.12.2021
Unique firms	189
Unique articles	144,383
Observations (firm-days)	87,198

Table 2: Horse race regressions — earnings calls

The following table presents the output from regressions of the form:

$$R_{jt} = \beta S_{jt} + \gamma X_{jt} + \varepsilon_{jt},$$

where t is the date of the earnings call; R_{jt} is the firm's buy-and-hold stock return minus the CRSP value-weighted buy-and-hold market index return over the 4-day event window, expressed as a percent; S_{jt} is one (or more) of our measures of sentiment, and X_{jt} are controls. Our controls include standarized unexpected earnings (SUE), log(book − market), log(size), log(shareturnover), industry fixed effects (Fama-French 49), a NASDAQ dummy and quarter-year fixed effects. For earnings calls prior to 2015, we train the MNIR model and extract which n-grams are annotated as positive and negative. The results presented in the table correspond to earnings calls from 2015–2020. We construct the sentiment measures using term frequency weights separately for LM positive/negative word lists (not included in the ML dictionaries), ML positive/negative dictionaries (not overlap with LM), positive/negative unigrams that overlap in the ML and LM dictionaries, and ML positive/negative bigrams. All sentiment measures are scaled to unit variance. Standard errors are clustered on FF49 industries and fiscal quarters. The table presents point estimates and t-statistics (in parenthesis).

	Dependent variable:					
	Filing period excess return					
	(1)	(2)	(3)	(4)	(5)	(6)
LM positive	0.41***				−0.14*	0.06
	(6.6)				(−1.9)	(1.0)
LM negative	−0.50***				0.39***	0.24***
	(−4.4)				(6.0)	(3.0)
ML positive		0.98***			0.78***	
		(7.7)			(8.7)	
ML negative		−1.37***			−0.94***	
		(−11.7)			(−9.8)	
LM & ML positive			1.25***		0.90***	0.89***
			(11.4)		(9.5)	(9.7)
LM & ML negative			−1.56***		−1.32***	−1.34***
			(−9.3)		(−9.1)	(−9.4)
ML positive bigrams				1.38***		1.06***
				(10.7)		(12.4)
ML negative bigrams				−1.36***		−0.79***
				(−7.7)		(−5.8)
Adjusted R2	0.021	0.046	0.054	0.045	0.065	0.064
Observations	39,269	39,269	39,269	39,269	39,269	39,269

Table 3: External validity — 10K filings

The following table replicates the analysis in Table 2, using the 10K filing as the event of interest, measuring sentiment using the LM and ML dictionaries. The dependent variable is the firm's buy-and-hold stock return minus the CRSP value-weighted buy-and-hold market index return over the 4-day event window. Our controls include log(book – market), log(size), log(shareturnover), industry fixed effects (Fama-French 49), a NASDAQ dummy and quarter-year fixed effects. The machine learning dictionaries are constructed using the earnings calls database, from 2006–2020, following the classifications in Table 2. We include all 10K filings, filtered as in Loughran and McDonald (2011), for the period 1995–2018. We construct the sentiment measures using term frequency weights separately for LM positive/negative word lists (not included in the ML dictionaries), ML positive/negative dictionaries (not overlap with LM), positive/negative unigrams that overlap in the ML and LM dictionaries, and ML positive/negative bigrams. All sentiment measures are scaled to unit variance. Standard errors are clustered on FF49 industries and fiscal quarters. The table presents point estimates and *t*-statistics (in parenthesis).

	Dependent variable:					
	Filing period excess return					
	(1)	(2)	(3)	(4)	(5)	(6)
LM positive	-0.14^{**}				-0.14^{**}	-0.13^{**}
	(-2.2)				(-2.1)	(-2.1)
LM negative	-0.06^{*}				0.03	0.01
	(-1.9)				(1.1)	(0.4)
ML positive		0.11^{***}			0.13^{***}	
		(5.1)			(4.9)	
ML negative		-0.05			-0.01	
		(-1.2)			(-0.3)	
LM & ML positive			0.05^{**}		0.05	0.05^{*}
			(2.4)		(1.5)	(1.8)
LM & ML negative			-0.18^{**}		-0.21^{***}	-0.14^{**}
			(-2.4)		(-2.9)	(-2.2)
ML positive bigrams				0.15^{***}		0.13^{***}
				(3.8)		(3.1)
ML negative bigrams				-0.16^{**}		-0.10^{**}
				(-2.6)		(-2.5)
Adjusted R2	0.013	0.013	0.013	0.013	0.013	0.014
Observations	76,922	76,922	76,922	76,922	76,922	76,922

Table 4: External validity — WSJ articles

The following table replicates the analysis in Table 2, using as the corpus of interest the publication of articles about a firm in the Wall Street Journal (WSJ). The dependent variable is the firm's buy-and-hold stock return minus the CRSP value-weighted buy-and-hold market index return over the 2-day event window around the article publication. Our controls include day-of-the-week, month and firm fixed effects. The machine learning dictionaries are constructed using the earnings calls database, from 2006–2020, following the classifications in Table 2. The WSJ database consists of 87,198 event days, corresponding to 189 unique firms, see Section 2.1 and Table 1 for details. We construct the sentiment measures using term frequency weights separately for LM positive/negative word lists (not included in the ML dictionaries), ML positive/negative dictionaries (not overlap with LM), positive/negative unigrams that overlap in the ML and LM dictionaries, and ML positive/negative bigrams. All sentiment measures are scaled to unit variance. Standard errors are clustered at the month and firm level. The table presents point estimates and t-statistics (in parenthesis).

	Dependent variable:					
	Filing period excess return					
	(1)	(2)	(3)	(4)	(5)	(6)
LM positive	0.10***				0.08***	0.09***
	(7.6)				(6.1)	(6.7)
LM negative	−0.11***				−0.05***	−0.08***
	(−7.5)				(−3.5)	(−5.8)
ML positive		0.12***			0.09***	
		(7.4)			(5.2)	
ML negative		−0.21***			−0.17***	
		(−10.4)			(−9.3)	
LM & ML positive			0.16***		0.12***	0.12***
			(10.7)		(8.8)	(9.2)
LM & ML negative			−0.19***		−0.16***	−0.16***
			(−8.7)		(−7.6)	(−7.5)
ML positive bigrams				0.14***		0.09***
				(7.2)		(4.9)
ML negative bigrams				−0.14***		−0.10***
				(−7.8)		(−6.0)
Adjusted R2	0.007	0.010	0.009	0.007	0.013	0.012
Observations	87,198	87,198	87,198	87,198	87,198	87,198

Table 5: Dictionary breadth

The following table presents the length of the different dictionaries considered in Section 3, as well as their breadth in the different corpora, measured by the percentage of the corpora that each dictionary covers.

		Coverage (% of corpus)		
Dictionary	Number of tokens	Earnings calls	10K	WSJ
Positive words				
LM positive	329	1.9%	0.7%	1.3%
ML positive	57	8.4%	4.1%	3.2%
ML & LM positive	18	1.3%	0.2%	0.4%
ML positive bigram	381	2.3%	0.8%	0.3%
Negative words				
LM negative	2,315	1.4%	2.7%	3.2%
ML negative	64	4.5%	4.3%	3.1%
ML & LM negative	30	0.4%	0.4%	0.5%
ML negative bigram	344	1.4%	0.7%	0.4%

Table 6: Top LM unigrams and ML scores

We consider the top 30 LM unigrams by frequency, separately for positive and negative words. For each of them, the table presents the total coverage (Cov., frequency over the whole earnings calls corpus measured in basis points), and the robust MNIR scores (positive and negative), namely the number of the 500 cross-validation samples for which that unigram is labelled as positive (negative) by the MNIR fit. Tokens colored in blue (red) belong to the ML positive (negative) unigram dictionaries.

Positive words				**Negative words**			
Token	Cov.	% Pos	% Neg	Token	Cov.	% Pos	% Neg
good^{+}	35.1	99.4	0.0	question	25.6	39.2	7.0
strong^{+}	26.0	100.0	0.0	questions	10.5	21.8	6.4
better^{+}	15.1	92.4	0.0	decline^{-}	8.0	0.0	99.8
opportunities	12.9	58.4	4.6	loss^{-}	6.8	0.0	99.0
able	12.1	63.2	2.2	negative^{-}	4.4	0.2	96.6
opportunity	11.9	68.0	3.8	difficult	3.7	0.0	78.4
positive	10.2	62.6	2.6	against	3.6	7.8	27.4
improvement^{+}	10.0	100.0	0.0	declined^{-}	3.5	0.2	91.4
progress	7.9	56.4	5.0	restructuring	3.2	30.8	30.4
pleased^{+}	7.7	99.8	0.0	losses	2.8	6.0	69.0
improved^{+}	6.9	100.0	0.0	challenges^{-}	2.6	0.0	99.8
improve	6.7	11.0	34.0	challenging^{-}	2.4	0.2	87.0
best	6.5	25.6	10.4	recall	1.8	8.2	25.6
strength^{+}	4.8	100.0	0.0	declines^{-}	1.8	0.0	85.8
success^{+}	4.4	88.8	0.0	volatility	1.7	6.8	42.4
excited	4.4	49.8	4.6	slow	1.6	0.2	66.4
profitability	4.3	63.0	4.8	break	1.5	22.6	6.6
confident^{-}	3.9	0.4	80.4	weakness^{-}	1.4	0.0	99.8
improving^{+}	3.8	82.4	0.0	bad	1.3	6.0	44.4
favorable^{+}	3.6	86.4	0.0	challenge	1.3	0.2	77.4
improvements^{+}	3.5	89.4	0.2	problem	1.3	1.6	71.4
gain	3.4	64.0	1.2	weak	1.2	0.2	78.8
despite	3.3	3.8	33.6	claims	1.2	12.0	61.8
successful	3.2	41.2	2.4	slower^{-}	1.2	0.0	93.0
gains^{+}	3.2	82.4	0.0	negatively^{-}	1.2	0.0	96.8
stronger	3.2	72.0	0.2	lost^{-}	1.2	0.0	96.8
efficiency	3.1	68.6	1.6	cut	1.1	3.4	50.2
advantage	3.0	61.0	1.4	slowdown^{-}	1.1	0.0	96.8
achieve	3.0	32.0	6.0	impairment	1.1	1.2	81.0
innovation	2.8	57.2	6.4	missed	1.0	0.6	49.2

Table 7: Top ML unigrams by frequency

We consider the top 30 ML unigrams by frequency, separately for positive and negative words. For each of them, the table presents the total coverage (Cov., frequency over the whole earnings calls corpus measured in basis points), and the robust MNIR scores (positive and negative), namely the number of the 500 cross-validation samples for which that unigram is labelled as positive (negative) by the MNIR fit. Tokens colored in blue (red) belong to the LM positive (negative) unigram dictionaries.

Positive words				Negative words			
Token	Cov.	% Pos	% Neg	Token	Cov.	% Pos	% Neg
think	97.0	82.4	2.0	not	108.5	0.6	97.0
growth	62.2	96.2	0.4	down	27.9	0.0	94.8
up	54.8	91.4	0.6	back	25.3	0.0	95.8
well	50.0	82.8	0.4	impact	21.8	0.0	99.8
over	47.3	87.8	0.2	believe	18.4	0.4	84.2
really	38.6	97.2	0.0	lower	17.3	0.0	100.0
continue	37.8	96.6	0.0	due	15.2	0.0	91.0
good^{+}	35.1	99.4	0.0	costs	14.1	0.4	89.2
results	27.9	91.2	0.0	expected	11.5	0.0	97.4
share	27.8	91.2	0.8	related	11.2	0.0	99.2
cash	26.9	83.2	1.6	change	10.3	0.0	96.2
increase	26.6	95.8	0.0	need	9.8	0.4	81.4
strong^{+}	26.0	100.0	0.0	offset	8.3	0.0	90.0
basis	25.5	84.0	1.4	expectations	8.0	0.2	81.2
operating	25.3	89.4	0.6	decline^{-}	8.0	0.0	99.8
margin	25.2	93.2	0.2	trying	7.3	0.4	83.2
lot	23.6	87.8	1.0	changes	7.0	0.0	95.0
years	20.9	81.4	0.4	loss^{-}	6.8	0.0	99.0
increased	20.4	96.4	0.0	term	6.7	0.4	83.6
income	17.5	88.6	0.6	certain	6.6	0.0	82.0
performance	17.2	94.0	0.0	factors	6.4	0.0	80.8
better^{+}	15.1	92.4	0.0	taking	6.0	0.0	96.0
pretty	14.3	94.0	0.0	understand	5.9	0.0	100.0
great	13.4	100.0	0.0	timing	5.7	0.0	97.8
across	12.2	91.4	0.2	however	5.2	0.0	99.8
continued	12.2	91.2	0.0	associated	4.9	0.2	91.2
flow	12.1	84.0	1.6	impacted	4.5	0.0	100.0
improvement^{+}	10.0	100.0	0.0	negative^{-}	4.4	0.2	96.6
benefit	8.9	83.6	0.2	decrease	4.4	0.0	96.2
pleased^{+}	7.7	99.8	0.0	issues	4.1	0.0	100.0

Table 8: Disambiguating unigrams

This table presents a subset of the bigrams associated with the six tokens *improve*, *confident*, *solid*, *soft*, *cash*, and *continue*. The column "Bigram" lists the bigram. The column "Rel. Freq." is the relative counts of the bigram out of all the bigrams that contain the given unigram i.e. 8.88% of the times "improve" is written, it is within the bigram "continue improve." The D^+ and D^- score columns refer to the robust MNIR scores, namely the proportion of the cross-validation subsamples where the *n*-gram is classified as positive (negative) minus the number when it is classified as negative (positive). Tokens marked with a positive/negative sign denote LM positive/negative words. Only bigrams with D^+ or D^- scores above 20% are presented.

Positive bigrams			Negative bigrams		
Bigram	Rel. Freq.	D^+ score	Bigram	Rel. Freq.	D^- score
continue$^+$ improve$^+$	8.88	64.4	improve$^+$ performance$^+$	2.17	39.6
continues$^+$ improve$^+$	2.91	41.6	improve$^+$ over$^+$	1.38	21.4
able$^+$ improve$^+$	0.97	22.8	going improve$^+$	1.11	25.2
			improve$^+$ second	0.80	27.6
			conditions improve$^+$	0.79	22.2
more confident$^+$	5.05	31.6	remain confident$^+$	15.82	75.2
increasingly confident$^+$	0.76	29.4	still confident$^+$	1.74	36.0
			confident$^+$ get	1.53	34.0
			confident$^+$ strategy	1.46	45.4
			confident$^+$ see	1.03	36.2
solid$^+$ growth$^+$	7.10	21.2	solid$^+$ tumors	1.19	28.6
solid$^+$ quarter	5.75	37.0			
solid$^+$ results$^+$	3.55	22.4			
quarter solid$^+$	2.53	21.8			
pretty$^+$ solid$^+$	2.51	23.4			
soft$^-$ launch	6.48	21.8	bit soft$^-$	10.62	25.0
			soft$^-$ demand	9.94	33.6
			soft$^-$ quarter	9.43	26.0
cash$^+$ flow$^+$	25.86	77.2	cash$^+$ used	0.54	42.8
free cash$^+$	10.58	63.0	cash$^+$ burn	0.33	27.6
operating$^+$ cash$^+$	2.89	42.4	cash$^+$ cost	0.32	39.2
cash$^+$ balance	1.96	37.6	company cash$^+$	0.26	26.8
strong$^+$ cash$^+$	1.45	83.2	used cash$^+$	0.23	23.4
continue$^+$ see	4.95	47.0	continue$^+$ believe$^-$	1.19	43.6
going continue$^+$	3.15	26.0	continue$^+$ advance	0.19	40.0
continue$^+$ grow	2.96	45.0	continue$^+$ face	0.14	36.8
expect continue$^+$	2.62	51.6	continue$^+$ impact$^-$	0.12	53.0
continue$^+$ focus	1.67	27.6	may continue$^+$	0.11	21.8

Table 9: Disambiguation: bigrams starting with “not”

This table presents a subset of the bigrams that start with the token “not,” a total of 1,267 unique bigrams (using our dtm with 2^{17} terms). The column “Term” lists the bigram. The column “Rel. Freq” is the relative counts of the bigram out of all the bigrams that start with “not,” i.e. 3% of the times “not” is written, it is within the bigram “not really”. Tokens marked with a positive/negative sign (blue/red) denote LM positive/negative words. We also mark with a positive/negative sign (cyan/brown) all the ML words.

Positive			Negative		
Term	Rel. Freq.	D^+ score	Term	Rel. Freq.	D^- score
not really$^+$	3.15	14.8	not think$^+$	4.53	43.4
not break$^-$	0.23	27.4	not going	4.05	53.4
not call	0.19	14.4	not get	1.39	53.4
not mind	0.18	16.4	not seeing	0.91	48.0
not undertake	0.18	13.4	not able$^+$	0.75	74.2
not guarantees	0.17	15.4	not changed$^-$	0.69	42.2
not surprised	0.14	17.0	not believe$^-$	0.63	53.0
not whole	0.13	22.6	not come	0.35	54.0
not spend	0.10	17.4	not enough	0.32	72.2
not sustainable$^+$	0.08	28.2	not getting	0.32	56.8
not single	0.06	20.2	not happen	0.30	49.2
not hard	0.06	26.2	not expected$^-$	0.17	60.0
not built	0.05	14.2	not issue$^-$	0.16	45.8
not thing	0.05	14.6	not meet	0.14	89.8
not statements	0.04	13.8	not affect	0.14	41.0
not driving$^+$	0.04	15.8	not related$^-$	0.10	47.0
not raise	0.04	19.0	not happy$^+$	0.10	79.6
not traditional	0.03	16.2	not lost$^-$	0.09	59.4
not happier	0.03	15.6	not satisfied$^+$	0.09	54.0
not actively	0.03	15.0	not performing	0.06	44.6
not tremendous$^+$	0.03	30.4	not materialize	0.06	61.0
not asked	0.02	18.2	not losing$^-$	0.05	73.8
not obvious	0.02	17.8	not perform	0.05	43.2
not separate	0.02	13.4	not offset$^-$	0.04	42.8
not raising$^+$	0.02	22.0	not pleased$^+$	0.04	51.0

Appendix

Parsing algorithm

The transcripts from the earnings calls are parsed and each paragraph is mapped to the manager, analyst, or operator speaking. Comments by the operator are subsequently removed. The earnings calls typically consist of two parts: an Introduction, typically scripted and read by the management team, and a Questions and Answers (Q&A) section where participants in the call can ask management about details of the earnings release. While we can separate the Introduction and the Q&A section of the call, we merge both parts.

Before proceeding to the creation of our new dictionaries, we perform a set of standard cleaning procedures from the NLP literature. We first remove non-ASCII characters and single character words. We split the strings into sentences and tokenize it, tagging each token using the NLTK package. We remove all words that are tagged as proper nouns by the NLTK tagger (codes NNP or NNPS), and other words such as determinants.[40]. We convert abbreviations to their full English word.[41] We eliminate all number characters, punctuation, and anything that are not alphanumeric characters. We remove stopwords starting with the list from the Snowball project in different languages.[42] We include/exclude a handful of terms into this stopword list.[43] Since one of our goals is to compare ML and LM word-by-word, and the LM dictionaries are unstemmed, we will present our results using unstemmed words.[44] We note that we are keeping the tokens no/not, which will have some bite when using bigrams regarding potential negation of positive words.

Data descriptions

These are the variables that we use in our tests:

1. Event period excess return: firm's buy-and-hold stock return minus the CRSP value-weighted buy-and-hold market index return over the 4-day event window, expressed as a percent.

2. Size: the number of shares outstanding times the price of the stock as reported by CRSP on the day before the event date.

3. Book-to-market: Book value is derived from quarterly Compustat for earnings calls and annual Compustat for 10-Ks. We derive book value as specified in Fama and French

[40] To be precise, we drop the following POS: NNP, NNPS, DT, SYM, CD, TO, LS, PRP, PRP$.

[41] This simply involves changing n't/not, 'll/will, 're/are, 'd/would, 'm/am, 've/have. We also change cannot/can not, as can is one of the stopwords we remove.

[42] Obtained from `http://svn.tartarus.org/snowball/trunk/website/algorithms/*/stop.txt`.

[43] We include the following words in our analysis that are part of the Snowball stopword list: against, above, below, up, down, over, under, again, further, few, more, most, no, not. We add can, will, must, and let. We also exclude all 2-character terms with the exception of no, up, and go.

[44] Previous versions of the paper constructed all the ML analysis using document-term-matrices (dtms) with stemmed words. The results using stemmed words are slightly stronger for the ML algorithm, but they penalize LM by "mis-stemming". For example both the words *quitting* and *quite* become *quit* when stemmed, which loses semantic meaning.

(2001) except for items not covered in quarterly Compustat. Market value is the number of shares outstanding times the price of the stock at the end of the last calendar year before the event date. We eliminate observations with a negative book-to-market.

4. Share turnover: The volume of shares traded in days $[-252, -6]$ prior to event date divided by shares outstanding on the event date. At least 60 observations of daily returns must be available to be included in the sample.

5. Pre FFAlpha: The Fama-French alpha based on a regression of their three-factor models using days $[-252, -6]$ relative to the event date. At least 60 observations of daily returns must be available to be included in the sample.

6. NASDAQ dummy: A dummy variable set equal to one for firms whose shares are listed on the NASDAQ stock exchange, else zero.

7. Standardized Unexpected Earnings (SUE): Unexpected earnings is computed as the difference between quarterly earnings per share (Compustat item EPSPXQ) minus earnings per shares from four quarters ago. SUE is defined as unexpected earnings scaled by individual firm's standard deviation.

The data depository[45] contains many other details, from the lists of uni/bigrams used in the the paper, to code that refines the dictionaries and replicates our analysis using public data.

ML dictionaries

LM&ML positive unigrams: achieved, benefited, benefiting, better, excellent, fantastic, favorable, gains, good, impressive, improved, improvement, improvements, improving, pleased, strength, strong, success.

LM&ML negative unigrams: challenges, challenging, decline, declined, declines, delay, delayed, delays, disappointed, disappointing, disappointment, inefficiencies, lack, losing, loss, lost, miss, negative, negatively, shortfall, slowdown, slowed, slower, slowing, underperformance, unexpected, unfortunately, weaker, weakness, worse.

ML positive unigrams: above, across, basis, benefit, cash, congrats, congratulations, continue, continued, continues, curious, delivered, driving, drove, exceeded, exceeding, expansion, flow, generated, great, grew, growing, growth, helped, helping, income, increase, increased, increasing, job, leverage, lot, margin, momentum, nice, nicely, operating, outperformance, outstanding, over, performance, pretty, proud, raising, really, record, repurchase, results, share, solid, sustainable, terrific, think, up, upside, well, years.

ML negative unigrams: actions, address, affected, affecting, anticipated, associated, back, believe, below, caused, causing, certain, change, changed, changes, confident, costs, decision, decrease, decreased, down, due, dynamics, expectations, expected, experienced, factors, fell, goodwill, happened, headwinds, however, impact, impacted, impacting, impacts, issue, issues, longer, lower, necessary, need, not, offset, pressure, pressures, pronounced, pushed, related,

[45] See `http://leeds-faculty.colorado.edu/garcia/data.html`.

resolve, revised, short, slipped, soft, softer, softness, steps, taking, temporary, term, timing, transition, trying, understand.

ML positive bigrams: able reduce, above expectations, above guidance, above high, above top, achieved record, across board, add congratulations, addition strong, aerospace defense, ahead expectations, ahead guidance, also benefited, also exceeded, also raising, balance sheet, based strong, basis point, basis points, beginning see, better anticipated, better expected, better guidance, came above, capacity utilization, capital management, cash flow, cash generation, certainly pleased, compares favorably, congrats again, congrats good, congrats great, congrats quarter, congrats strong, congratulations again, congratulations good, congratulations great, congratulations quarter, congratulations strong, consecutive quarter, continue grow, continue improve, continue see, continued focus, continued growth, continued improvement, continued momentum, continued strong, continuous improvement, couple years, data centers, delivered outstanding, delivered record, delivered strong, demand across, deposit growth, deposit side, diluted earnings, diluted share, done great, driven improved, driven improvement, driven record, driven strong, driving growth, drove strong, due strong, earnings per, efficiency ratio, end market, end markets, even better, exceeded expectations, exceeded guidance, exceeded high, exceeding guidance, exceeding high, excellent execution, excellent job, excellent quarter, excellent start, exceptional quarter, exceptionally well, executed well, executing well, execution across, execution team, existing customers, expanded basis, expect continue, expense leverage, expense management, extra week, extremely pleased, fantastic quarter, favorable mix, favorable product, first congratulations, flow generation, flow quarter, free cash, generated free, given strength, given strong, good execution, good job, good momentum, good quarter, good results, good see, great execution, great hear, great job, great quarter, great results, great see, great start, grew over, growth across, growth driven, growth margin, growth quarter, growth seeing, hard work, helped drive, helping drive, high end, higher gross, higher sales, hitting cylinders, home equity, impressive quarter, improved basis, improved gross, improved margins, improved operating, improved operational, improved outlook, improved performance, improved profitability, improved significantly, improvement across, improvement basis, improvement compared, improvement driven, improvement gross, improvement operating, improvement over, improvement quarter, income increase, income increased, income per, income quarter, income up, increase adjusted, increase basis, increase compared, increase gross, increase guidance, increase net, increase over, increase prior, increased basis, increased guidance, increased per, increased revenue, increased sequentially, increasing full, increasing guidance, inflection point, job quarter, just curious, just great, just talk, last quarter, life sciences, linked quarter, loan growth, loan portfolio, lot more, margin expanded, margin expansion, margin improved, margin improvement, margin improvements, margin increased, margin performance, margin up, margins improved, mentioned pleased, merchandise margin, momentum business, momentum going, momentum seeing, more efficient, more impressive, more more, net debt, net income, net interest, new customers, nice improvement, nice job, nice quarter, nice results, nice see, obviously great, obviously nice, obviously pleased, obviously strong, okay great, operating income, operating leverage, operating margin, operating margins, operating ratio, organic growth, outstanding performance, outstanding quarter, outstanding results, over last, over prior, overall pleased, particular strength, particularly pleased, particularly strong, pay down, payment volume, per diluted, per share, per square, percentage revenue, percentage sales, performance across, performance driven, performance exceeded, performance quarter, pleased financial, pleased first, pleased performance, pleased quarter, pleased report, pleased results, pleased second, pleased see, pleased strong, pleased third, point improvement, positive momentum, pretty good, pretty impressive, pretty much, pretty strong, product gross, quarter exceeded, quarter good, quarter great, quarter improved, quarter improvement, quarter increase, quarter increased, quarter nice, quarter performance, quarter record, quarter strong, quarter up, raise guidance, raised guidance, raising full, raising guidance, raising revenue, range up, really good, really great, really happy, really helped, really impressive, really nice, really pleased, really starting, really strong, really well, record adjusted, record compared, record earnings, record gross, record net, record operating, record quarter, record quarterly, record results, record revenue, record revenues, record up, report strong, reported record, repurchase program, result strong, results demonstrate, results driven, results exceeded, results strong, revenue exceeded, revenue grew, revenue growth, revenue increased, revenue up, sales increase, sales up, sales wholesale, saw nice, seeing benefits, seeing strength, sequential growth, sequential improvement, sequential increase, share above, share count, share gains, share increase, share increased, share repurchase, share up, share well, significant improvement, solid execution, square foot, starting see, still lot, strength across, strength business, strength quarter, strength saw, strength seeing, strong across, strong cash, strong demand, strong execution, strong financial, strong finish, strong first, strong fourth, strong growth, strong momentum, strong operating, strong performance, strong quarter, strong results, strong revenue, strong second, strong sequential, strong start, strong third, strong top, stronger anticipated, stronger expected, strongest quarter, summary pleased, sustainable going, taking share, talk little, tax rate, team done, terrific quarter, think continue, think sustainable, third consecutive, up basis, up compared, up over, up prior, up sequentially, upside quarter, wanted ask, well above, well ahead, well favorable, year raising, year strong.

ML negative bigrams: actions address, actions taking, additional cost, address issues, adversely impacted, aggressive pricing, also affected, also impact, also impacted, also impacting, also incurred, also negatively, arms around, average day, back half, back track, back up, based upon, beat dead, became clear, believe prudent, believe right, below expectation, below expectations, below expected, below guidance, below midpoint, biggest impact, bit longer, came below, cell lung, challenges business, challenges quarter, challenging quarter, change guidance, changes making, come back, compared positive, competitive pressure, competitive pressures, confidence not, confident strategy, continue impact, corrective actions, cost overruns, cost related, costs associated, costs increased, costs related, current challenges, customer orders, day rates, dead horse, decline adjusted, decline driven, decline due, decline gross, decline primarily, decline quarter, decline revenue, decline revenues, decline sales, declined quarter, decrease compared, decrease gross, decrease revenue, despite challenges, disappointed results, discussed earlier, down basis, down constant, down prior, driven lower, due decrease, due delays, due lower, due primarily, excess inventory, execution issues, expect begin, expenses related, factors impacted, fall short, fell short, felt like, first quarter, generic products, get back, get done, get worse, gives confidence, go back, going take, good news, goodwill impairment, growing pains, guess just, guidance down, guide down, half year, happened quarter, help understand, higher cost, higher costs, higher labor, however believe, impact coronavirus, impact lower, impact not, impact quarter, impact revenue, impact third, impacted ability, impacted first, impacted lower, impacted quarter, impacted results, impacted revenue, impacted third, impacting revenue, impairment charge, impairment charges, income decreased, increase cost, increase inventory, increased competition, increased cost, increased promotional, incremental costs, inventory adjustments, inventory correction, issue not, issue quarter, issues not, issues quarter, just matter, just not, just trying, labor costs, lack visibility, large customer, late quarter, leadership changes, legacy business, line expectations, line guidance, long term, longer anticipated, longer expected, longer sales, losing market, losing share, loss compared, loss continuing, loss first, loss per, loss primarily, loss quarter, loss third, lost business, lost revenue, low end, lower anticipated, lower demand, lower earnings, lower end, lower expectations, lower expected, lower gross, lower guidance, lower incentive, lower margin, lower margins, lower net, lower revenue, lower revenues, lower sales, lower volume, lower volumes, made decision, make changes, make sure, make up, margin compression, margin decline, margin declined, margin decreased, margin down, margin pressure, margin pressures, margins down, margins impacted, market dynamics, meet expectations, mix issue, mixed results, months ended, more aggressive, more cautious, more challenging, more competitive, more conservative, more pressure, more promotional, more pronounced, most significant, near term, need make, negative impact, negatively impacted, net loss, not able, not believe, not come, not enough, not expected, not get, not getting, not going, not happen, not happy, not issue, not losing, not lost, not materialize, not meet, not pleased, not related, not satisfied, not seeing, operating loss, operating losses, operational challenges, operational issues, originally anticipated, part challenge, part issue, partially offset, perfect storm, performance issues, positive note, price pressure, price reductions, pricing pressure, primarily due, product transition, profit decreased, profit down, project delays, promotional activity, quarter challenging, quarter decline, quarter decrease, quarter decreased, quarter down, quarter fell, quarter impacted, quarter loss, quarter not, quickly possible, reduction revenue, related acquisition, remain confident, remains intact, result lower, resulting lower, results below, results lower, return growth, revenue decline, revenue declined, revenue decrease, revenue decreased, revenue down, revenue expectations, revenue impacted, revenue outlook, revenue recognition, revenue shortfall, revenues declined, revenues decreased, revenues down, revised guidance, revised outlook, right decision, right thing, sales decline, sales down, sales force, saw slowdown, second half, sense urgency, sequential decline, several factors, share loss, short expectations, short term, significant decline, significant impact, significantly impacted, slightly below, slow start, slowed down, slower anticipated, slower expected, slower start, softer expected, stable disease, step back, step down, still believe, still feel, stock down, strategic review, student starts, take little, take longer, take time, taken actions, taken longer, takes time, taking actions, taking longer, taking necessary, taking question, taking steps, thought going, timing issue, tough quarter, traffic trends, trying reconcile, trying understand, under pressure, understand not, used operating, used operations, vessel revenue, want make, weaker expected, weakness saw, weakness seeing, wind market, within guidance, worse expected.

Table 10: Horse race regressions— earnings calls

The following table corresponds to Table 2 including the coefficients on all controls, and adding a column for the specification without any sentiment variables.

	Dependent variable:						
	Filing period excess return						
	(1)	(2)	(3)	(4)	(5)	(6)	(7)
LM positive		0.41***				−0.14*	0.06
		(6.6)				(−1.9)	(1.0)
LM negative		−0.50***				0.39***	0.24***
		(−4.4)				(6.0)	(3.0)
ML positive			0.98***			0.78***	
			(7.7)			(8.7)	
ML negative			−1.37***			−0.94***	
			(−11.7)			(−9.8)	
LM & ML positive				1.25***		0.90***	0.89***
				(11.4)		(9.5)	(9.7)
LM & ML negative				−1.56***		−1.32***	−1.34***
				(−9.3)		(−9.1)	(−9.4)
ML positive bigrams					1.38***		1.06***
					(10.7)		(12.4)
ML negative bigrams					−1.36***		−0.79***
					(−7.7)		(−5.8)
log(Size)	−0.03	−0.03	−0.13*	−0.20***	−0.20***	−0.25***	−0.28***
	(−0.5)	(−0.5)	(−2.1)	(−2.9)	(−3.2)	(−3.3)	(−3.8)
log(Book-to-Market)	−0.03	0.00	0.09	0.05	0.06	0.10	0.08
	(−0.3)	(0.0)	(0.8)	(0.5)	(0.5)	(0.9)	(0.7)
log(Share turnover)	−0.12	−0.12	0.01	−0.13	0.05	−0.05	−0.01
	(−0.9)	(−0.9)	(0.0)	(−1.0)	(0.4)	(−0.3)	(−0.1)
SUE	0.96***	0.93***	0.82***	0.79***	0.81***	0.75***	0.73***
	(6.9)	(6.9)	(7.1)	(7.3)	(7.0)	(7.4)	(7.3)
Pre FFAlpha	−2.67***	−2.91***	−4.48***	−4.52***	−4.18***	−5.29***	−5.10***
	(−3.6)	(−3.9)	(−5.5)	(−5.3)	(−4.8)	(−5.7)	(−5.3)
Nasdaq dummy	0.11	0.10	0.11	0.14	0.10	0.13	0.13
	(0.7)	(0.7)	(0.8)	(0.9)	(0.7)	(0.9)	(0.8)
Adjusted R2	0.017	0.021	0.046	0.054	0.045	0.065	0.064
Observations	39,269	39,269	39,269	39,269	39,269	39,269	39,269