The color of finance words*

Diego García† Xiaowen Hu‡ Maximilian Rohrer§

December 9, 2021

Abstract

We study a standard machine learning algorithm (Taddy, 2013) to measure sentiment in financial documents. Our empirical approach relies on stock price reactions to color words, providing as output dictionaries of positive and negative words. 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.

*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. 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 it 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 statement.

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 ideas behind other machine learning algorithms in the literature are likely to produce similar (or

[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.

better) results.[3]

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 show that bigrams perform significantly better than unigrams, 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 using bigrams, we note that a baseline specification of the stock price reaction to the earnings call event with controls has an R^2 of 1.9%, which the LM dictionaries raise to 4.5%. Our simplest unigram specification has an R^2 of 4.6%, whereas using bigrams it goes up to 5.8%. When refining our dictionaries requiring stability across different cross-sections/time-periods, which we refer to as ML "plain money English" dictionaries, the R^2 of our out-of-sample regressions are all above 6%.[4]

We also study the external validity of the new dictionaries, and existing ones, across 10-K statements. We ask whether the ML dictionaries constructed using the earnings call corpus can predict price reactions to 10-K filings, and vice versa. The ML dictionaries generated using earnings calls are much more informative than the LM dictionaries in the context of 10-K releases. We also find that the dictionaries constructed using earnings calls are much more informative than dictionaries that stem from the 10-K filing date price reactions. This is probably not too surprising, since the timing of the earnings calls precedes the 10-K filings.

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). We calibrate the exercise so we end up with a few hundred unigrams and bigrams, which we consider one of the main outputs of

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

[4]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 can be applied in different contexts/documents.

our research agenda.[5] These "plain money English" dictionaries perform excellent relative to the LM dictionaries using both samples of 10-K releases and earnings calls, the two corpora we study in our paper.

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).[6] 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.[7]

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

[6]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).

[7]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).

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,[8] 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.[9] 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 predictability results, where we compare the performance of the different dictionaries in the context of earnings call transcripts. Section 4 studies dictionary breadth, the LM words in more detail, and discusses the disambiguation that our bigram representation achieves. In Section 5 we study 10-K statement releases to assess the external validity of the new dictionaries. Section 6 looks at the stability of the estimated dictionaries across time and industry cross-sections, and it discusses our "plain money English" dictionaries. The Appendix includes further details.

2 Measuring sentiment

In this section we first discuss the financial text corpus 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 our method for constructing new dictionaries. We end the section by outlining our empirical approach.

[8] 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.

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

2.1 Earnings calls corpus

Our paper studies the corpus from earnings calls, namely the transcripts from the call between the firm's management and analysts/investors. 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.[10] 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.

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 2016. The second is the earnings calls transcripts as provided by Wall Street Horizons, which covers the period 2009–2019. The intersection of these two datasets over the overlapping period 2009–2016 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 60,599 observations consisting of 3,292 unique firms.

The transcripts are subsequently 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 question and answer section of the call, for our main analysis we merge both parts.[12]

[10] See Section 5.1 for more details on the stock price reactions to earnings calls versus 10-K releases.

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

[12] See Table 12 in the Appendix for analysis separating the Introduction and the Q&A sections of the earnings call.

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.[13]. We convert abbreviations to their full English word.[14] 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.[15] We include/exclude a handful of terms into this stopword list.[16] 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.[17] We note that we are keeping the tokens no/not, which will have some bite when using bigrams regarding potential negation of positive words.

Given the above parsing procedure, 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 number 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.[18] 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.[19]

It is important to put the above number in the context of the size of the document-term-matrices (dtm) we will be constructing below. 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

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

[14]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.

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

[16]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.

[17]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.

[18]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).

[19]The number of unique unigrams is 186,994. It is worthwhile noticing these numbers are "large," as standard estimates of English native speakers dictionaries range around the 20,000 word limit. 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.

100, despite all the cleaning/token removal we have performed this far. We further discuss the breadth of such text representations in Section 4.1.

2.2 Multinomial inverse regression

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 earnings call 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 tokens, out of a set of p total n-grams, appear in each earnings call. We will use a dtm with 65,536 (2^{16}) n-grams in our baseline specifications, using the most frequent n-grams. 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 and trigrams we are keeping some context, which will prove crucial when disambiguating words.

The multinomial inverse regression (MNIR) model has a Bayesian flavor, belonging to a class of algorithms close to topic models (such as LDA).[20] 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 words (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

[20]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).

of dimensionality in typical machine learning fashion,[21] (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.[22] These loadings are roughly evenly distributed into positive/neutral/negative in our baseline specifications. We classify the n-grams into two dictionaries: one consisting of n-grams with positive loadings, one consisting of those that have negative loadings. We will refer to these dictionaries as ML dictionaries in what follows, as they are constructed using the output from the ML algorithm.

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 convolution layer in our algorithm, with a similar flavor to a lasso penalty: reduce the dimensionality of the final sentiment representation.[23]

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.[24] We conjecture that using more modern methods will only widen the "machines versus humans" divide we document using the MNIR algorithm.

2.3 Quantifying sentiment

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

[21] 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).

[22] 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.

[23] See Section 6.2 for further discussion on how to refine the dictionary construction. Table 13 in the Appendix, as well as the evidence in Table 10, shows that there is little cost in dropping the information in the MNIR coefficients.

[24] 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.

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 354 positive and 2,355 negative terms.[25]

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.[26]

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 65,536 (2^{16}) most frequent n-grams in our baseline specifications.

A (positive/negative) dictionary is a set of words along the m words in a given dtm, $\mathscr{D}_i$, a subset of the vocabulary $\mathscr{V}$. We can represent this as matrix D_i of the same column dimension as the dtm under consideration, with each row 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

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

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 (and trigrams) we can capture more nuanced aspects of the English language.

[25] We use the version of the Loughran and McDonald (2011) dictionaries as shared by the authors in their webpage as of 2017. We use the GI dictionaries as supplied by the `SentimentAnalysis` package in R.

[26] 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.

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, bigrams, and trigrams, we will have different sets of dictionaries developed by the machine learning algorithm. One of the goals of our paper is to compare the predictability, out of sample, of such LM and ML dictionaries, to which we turn next.

2.4 Empirical approach

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 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 (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 both the sentiment variables and the controls at 1/99%. 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).[27] These controls complement the inclusion of hard

[27] 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

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.

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 earnings calls prior to January 1st, 2015, a total of 32,955 events (2006–2014 subsample). As the out-of-sample dataset, we use all the earnings calls on or after January 1st, 2015, 27,644 events (2015–2019 subsample). 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 sketched above is not critical for our results. We could sample particular time periods, or do 80/20 training/validation, and our qualitative and quantitative results are very similar.

3 Predicting returns using text

In Section 3.1 we first consider creating dictionaries using the machine learning algorithm described in Section 2, and we study whether such classification has bite for predicting stock price reactions. The main predictability results of the paper are discussed in Section 3.2, where we compare the performance of the machine learning algorithm to that from the standard bag-of-words approach (Loughran and McDonald, 2011).

3.1 Preliminary results

The goal of this section is modest: we simply ask whether the machine learning algorithm can indeed pick up positive and negative sentiment text, and whether uni/bi/trigrams perform better. We first implement the machine learning algorithm in our earnings calls corpus using unigrams, which is the closest to the bag-of-words approach in Loughran and McDonald (2011). We then look into bigrams and trigrams to gather to what extent such high-dimensional representations of the underlying text help in predicting stock return movements, as suggested in the NLP literature.

the year than 10-K releases, which makes the benefits of Fama-MacBeth more muted.

The analysis in this section is completely unsupervised by humans: we are going to let the machine learning algorithm figure out which of the 65,536 (2^{16}) n-grams under consideration have predictive power, using stock price reactions as a guide. This is in sharp contrast to bag-of-words approaches, in which researchers are asked to label the color of a given word. We start with the top 65K n-grams by frequency, a parameter which we will revisit in Section 4.1.

Following the algorithm described in Section 2.4, we fit our model using the earnings calls from 2006–2014. The fitted MNIR object is then used to create the positive and negative dictionaries, depending on the signs of the loadings in the machine learning algorithm. Armed with these dictionaries, we then try to predict stock price reactions on the held-out sample, namely the calls from 2015–2019.

In the first column in Table 1, we report the results using unigrams.[28] We find that both the positive and negative dictionaries constructed using the MNIR estimates significantly predict the stock market reactions out-of-sample. 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 0.71% increase (-1.64% decrease) in the stock price reaction. The R^2 of the regression increases from 1.9%, in a specification without any of the textual variables, up to 4.6% when including the two textual measures.

The second column in Table 1 repeats the exercise for bigrams. We see that the predictability is significantly stronger than when using unigrams, with absolute t-stats around 8, and an adjusted R^2 at 5.8%, 1.2% higher than with unigrams. The marginal effects are also stronger: a one standard deviation change in the positive (negative) sentiment scores result in increases (decreases) in the stock price reaction amounting to 1.73% (-1.55%).

The third column in Table 1 reports the estimates using the top 65K trigrams in our empirical exercise. The predictability is fairly strong, comparable to that of bigrams, with absolute t-stats above 7. The economic magnitudes are slightly smaller than in the case of bigrams, with a significantly lower R^2 (3.7%). This is likely explained by the sparcity of the trigram representation (see Section 4.1).

The last column in Table 1 compares the performance of the three sets of dictionaries (unigrams, bigrams and trigrams) jointly. The regression reported in this last column tries to tease out which of the text representations has more bite. We find that unigrams and trigrams do not seem to bring much to the table, relative to the bigrams. Only the bigram coefficients are statistically different from zero, with similar point estimates and R^2 to the case where we estimated only using bigrams.

[28] In Table 11 in the Appendix we include the results including all control variables.

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) with 65K tokens (ordered by frequency). 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 2015–2019. The algorithm labels all the 65K tokens as positive/neutral/negative, which allows us to construct sentiment dictionaries following the recipe in Section 2.2. We next compare such dictionaries to those from Loughran and McDonald (2011).

3.2 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 standard bag-of-words approach. Our main horserace will be against the sentiment metrics 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.

Table 2 compares the LM and ML sentiment scores. The first column presents the estimates of (2) using the LM dictionaries as a sentiment metric.[29] We find that both positive and negative sentiment have significant predictive power, with magnitudes similar to the unigrams results from the ML algorithm in Table 1, with a slightly lower R^2 (4.5% versus 4.6%). We highlight how the LM dictionaries actually 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.

Column two in Table 2 presents the first head-to-head regression among the two (LM/ML) sentiment metrics: it reports coefficients on both LM and ML unigram scores. Our estimates suggest that both unigram dictionaries, LM and ML, have significant explanatory power, raising the R^2 of our regression from 1.9% to 5.5% when jointly estimated (4.5% LM only, 4.6% ML only). While the t-stat for the LM negative is the largest in absolute value, all four dictionary scores are significant, with the ML coefficient magnitudes roughly the same than those from the LM words (0.64 versus 0.71 for positive sentiment, -1.00 versus -0.73 for negative sentiment).

The skeptical reader may conjecture that the ML dictionaries are simply picking up LM words. One of the advantages of working with unigrams is that it allows us to decompose the different

[29] We emphasize that we do not use the 65K token limit when computing the sentiment scores using the LM dictionaries.

dictionaries into those that belong to both the LM and ML, and those that only belong to either the LM or the ML lists. We do a careful comparison of both dictionaries in Section 4.2, but at this point we can simply eliminate any ML word that is contained in the LM dictionaries. We can then create our sentiment scores with this (smaller) dictionary that does not contain any LM words. Column three of Table 2 presents the horserace regression when the ML dictionaries do not overlap with the LM dictionaries. There are small changes in the expected direction: stronger effects for LM, slightly weaker for ML. But the fit is actually pretty similar, with the coefficients and statistical significance very close to those reported in column two. The new ML negative unigrams are comparable, in terms of their impact on stock returns, to the LM words.

The unigram representation was strongly dominated by bigrams in the analysis from Section 3.1 (see Table 1). In column 4 of Table 2 we repeat the horserace regressions using bigrams instead of unigrams. The results mimic the strong evidence in Table 1: the R^2 of the regression increases up to 6.4% when adding the ML sentiment scores, and the LM variables lose some of their predictability, as their coefficients are half the size compared to the standard alone specification in column one. The economic magnitude of the ML coefficients is 2–3 times bigger than the LM coefficients (1.39 versus 0.57, -1.22 versus -0.65).

The last column in Table 2 reconsiders the concern that the ML dictionaries may include LM words. We label a bigram as having overlap with LM if one of its member words are part of the LM dictionaries. We exclude any such bigram from the ML dictionaries and then compute sentiment scores, presenting the horserace regression in the last column of Table 2. The point estimates move as with unigrams, but the overall fit is qualitatively identical: the ML bigrams have marginal impacts of 1.0–1.2% per one standard deviation change, compared to 0.7–0.8% for the LM scores, with the bulk of the predictability, in terms of increase in R^2, coming from the ML dictionaries.

Overall, the empirical evidence in Tables 1–2 strongly advocates for the dictionaries brought up by the machine learning algorithm. The n-grams selected by the MNIR model outperform the standard LM dictionaries, both in terms of statistical significance (and goodness-of-fit), and the economic magnitudes of their impact on asset prices. This is particularly true for the bigram specification. We provide further color in the rest of the paper, trying to highlight why and how the machine learning algorithm does better than the human dictionaries.

4 Coloring words

At the heart of our ML approach is to measure sentiment using large document-term-matrices (dtms) that summarize the underlying text in the earnings calls. The results in Section 3 suggest that dtms with 65K terms 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.

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 65K n-grams 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.

One of the important differences between the LM and the ML approaches is with respect to the breadth of the dictionaries. When implementing the machine learning algorithm, our first step was to narrow the relevant corpus by including only the top 65K n-grams. The MNIR estimation will pick roughly 9K n-grams as positive, and a similar number as negative, with 47K n-grams labeled as neutral. We remind the reader that this is a larger number of elements than the unigrams in the Loughran and McDonald (2011) dictionaries: the LM negative word list consists of 2,355 tokens, and the LM positive word list only 354 tokens. This said, the 65K n-gram limit we imposed in the previous section is arbitrary, so we explore next to what extent it binds.

We address the breadth of our dictionaries by studying how different sized n-gram representations perform. At the heart of studying any textual corpus is the question of efficient/meaningful choices of terms to include. The tradeoff the ML algorithm tries to tackle is the very standard efficiency versus bias in non-parametric statistics: more n-grams means more signals (efficiency), at the cost of overfitting (bias).

The top panel of Figure 1 plots the percentage of the corpus that is covered by dtms with 2^k terms, for $k = 9, \ldots, 25$. 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 1 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 call 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 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.

In Figure 2 we plot the R^2 of a regression as in Table 1, training pre 2015, predicting 2015–2019, simply changing the size of the dtm we use. The figure considers dtms of size 2^k, for $k = 7, \ldots, 20$. The unigram specification peaks around 4.8%, using 4,096 tokens. For bigrams the R^2 is fairly flat at around 6% for dtms with more than 65K terms. For trigrams there is improvement as we increase the dtm up to 131K, but the R^2 are always significantly below those of bigrams, and it plateaus for larger dtms.

This evidence seems to suggest that the 65K token dtm we use in our base case is fairly well calibrated for uni/bi/trigrams. We could have used a lower number of terms for unigrams, and a higher number for bi/trigrams, with slightly better fits, but the 65K restriction seems like a good compromise.[30]

We note that given the evidence provided this far, it is natural to drop the trigram analysis, as it does not seem to provide any performance improvements over bigrams. Furthermore, it also seems reasonable to focus on smaller dtms for unigrams, as Figure 2 suggests the ML algorithm is overfitting with larger dtms, and Figure 1 showed that a dtm with 4096 terms (2^{12}) already covers more than 95% of the whole corpus.

[30] In the Appendix, Tables 14 and 15, we decompose the predictability of the ML dictionaries by frequency. The main conclusion is that the first quintile of the words (in terms of token frequency) does not bring much to the table relative to less frequent quintiles.

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. We note that the machine learning algorithm will create a different set of dictionaries as we change the training sample. For the purposes of this section, we use the same specification as in Section 3 (using earnings calls 2006–2014, as in Tables 1–2). We compare the n-grams that are positive/negative to those in the LM dictionaries. The analysis of unigrams maps one-to-one to a dictionary approach (as in Loughran and McDonald (2011)), while the analysis gets a bit more nuanced when moving to n-grams when $n \geq 2$, as we have to consider all n unigrams that compose a given n-gram.

Table 3 provides confusion matrices describing the overlap between the LM and the ML dictionaries. The first row in Panel A shows that out of the 354 positive LM terms, 116 of them are classified as ML positive, 44 are ML negative, 192 of them are neutral, and 2 are not part of the top 65K unigrams in the earnings calls corpus. In parenthesis we provide the term frequencies, over the whole earnings calls corpus, of each of the terms. We see that the LM positive words have a frequency of 3.5%, 2.8% of which overlaps with ML positive words, with only 0.2% of the word frequencies in disagreement between positive/negative (0.6% are considered neutral). The second row considers the LM negative dictionary. Out of the 2,355 LM negative words, 164 of them are classified by the ML algorithm as positive, 513 as negative, 1,242 as neutral, and 436 of the words do not make it into the top 65K unigrams. We note that there is more disagreement on the negative word lists between the LM and ML dictionaries: roughly half (1.0% of the total frequency of 1.9%) are consistently labeled as negative, with 0.2% being consider by the ML algorithm as positive, and 0.7% neutral.

Panel B lists all the ML bigrams, documenting how many of them have LM words that are positive/negative/both positive and negative, as well as how many ML bigrams do not contain any LM word. In parenthesis we present the term-frequencies corresponding to a given group of words. Out of the 8,655 bigrams labelled as positive by the ML algorithm, which correspond to 10.7% of the corpus, the vast majority, 7,163 terms (9.2% of the corpus) do not have any LM word. We see a similar lack of LM terms in the ML negative bigrams: out of the 9,517 bigrams labelled as negative, (9% of the corpus), 8,561 terms (8.3% of the corpus) do not have any LM word. There is some agreement on the positive side, where about 1,352 bigrams having a positive LM word, and only 134 having negative LM words (1.4% versus 0.1% in terms of frequencies). There is also some agreement on the negative side, with 593 bigrams having a negative LM word, but 357 having positive LM words (0.4% versus 0.2% in terms of frequencies). Despite this overlap, the evidence in Panel B points to an independent reading of

sentiment using the ML bigrams, consistent with out results in Table 2.

In Panel C, we document the number of signals in our baseline specification with 65K tokens, comparing it to the LM dictionaries in terms of token size ("Signals"), and the percentage of occurrences, their term frequency, over the whole corpus ("Coverage"). The LM positive/negative words cover about 3.5/1.9% of the earnings calls corpus, in line with previous studies.[31] The ML unigrams are much more greedy, covering over 42% of the corpus for positive sentiment, and 31% with negative sentiment. The MNIR model is "too inclusive" with unigrams, calling more than 70% of them either positive or negative, which may explain the weak results on unigrams in Table 2.[32] The bigram specification covers 9–11% of the corpus, also a larger chunk than the LM dictionaries. Note this is true despite the fact that the 65K bigram dtm we started with only covers about 50% of the full corpus. The MNIR model is fairly inclusive relative to the LM dictionaries.

The previous evidence argues that the ML algorithm is bringing many new terms to our sentiment measures, but it does not detail how it interacts with the LM words. In Table 4 we present the top 30 positive and negative words in the LM dictionaries by frequency, together with several statistics generated using the ML output. 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.[33] The table lists the token in consideration, its coverage over the whole corpus (Cov., measured in basis points), the total number of bigrams associated with that term (N_b), from a dtm with 65K bigrams, and the (frequency weighted) percentage of bigrams that are positive and negative according to the ML algorithm.

The ML algorithm broadly agrees with the LM classification. On the positive words, we find *good*, *strong*, *great*, and *improvement* that are mostly (50%+) classified as positive ML bigrams.[34] Similar agreements can be found in the negative domain for words such as *decline(d)*, *loss(es)* or *force* . There is some disagreement due to external validity, i.e. *question(s)* is a very special word in earnings calls. But the disagreement is a bit more nuanced: the ML algorithm

[31] We note that the number of unique negative LM words is smaller in Panel C than in Panel A. This is due to 291 LM terms not appearing in the earnings calls corpus. We emphasize that we include the full corpora in our LM sentiment scores, in particular the 145 (2) LM negative (positive) words that do not make it into the top 65K tokens in the earnings calls corpus).

[32] The regularization step we discuss in Section 6, which requires stability across different subsamples, goes a long way in solving this issue.

[33] This is not driven by differences in the earnings calls corpus and 10-K statements. Using 10-K statements, as in Section 5.1, 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.

[34] We note that we use the LM dictionaries as defined in 2017. The updated 2020 LM list excludes 20 terms, relative to the 2017 version. Among these 20 terms we have *great* and *benefit*. While the evidence in Table 4 supports the exclusion of *benefit*, it actually questions the exclusion of *great*, which shows up as a very positive sentiment word (see also Table 5).

does not consider *best*, *greater*, or *able* that positive, and it flags *confident* as a fairly negative word.[35] On the negative domain, *break* is labelled as positive by the ML algorithm, and other words such as *recall*, *critical*, and *closed/closing* are not particularly negative according to the ML bigram scores. Our approach captures color of finance discourse that is not measured by the standard bag-of-words approaches.

4.3 Disambiguation: bigrams versus unigrams

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. Each bigrams is associated to two unigrams, and we have estimated sentiment scores for each of these bigrams. We argue in this section that using the color of these bigrams helps understand the sentiment of individual unigrams, and that bigrams are extremely useful at disambiguating the meaning of words.

In order to visualize the disambiguation performed by the ML algorithm, we start with the fitted values from a dtm containing 65K bigrams. We then take the unique unigrams that are contained in these bigrams, a total of roughly four thousand unigrams.[36] For each of these unigrams, we compute the (frequency weighted) share of bigrams that are positive/neutral/negative. Figure 3 plots each of these unigram loadings as a ternary plot, with the neutral coordinate on top, negative/positive on the left/right.

Under the null hypothesis that there is no need for disambiguation, we would expect all points to concentrate in the three corners. 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 have a majority of neutral bigrams, with the rest of bigrams split evenly between positive and negative. There are terms that are not associated with any positive/negative bigrams, plotted on the sides of the triangle, but we see that the ML algorithm classifies many such terms as neutral.

We note that there are terms that seem to be fairly unambiguous, but most of these unigrams are not part of the LM word lists (plotted in red/blue). Figure 3 shows that the LM positive/negative words do cluster on the right/left sides of the ternary plot, but the majority of them have neutral

[35] Out of all the bigrams that contain *confident* that occur more than 1,000 times, the following are considered negative: *remain confident*, *feel confident*, *confident ability*, *confident going*, *highly confident*, *still confident*. Only two bigrams *more confident* and *confident continue* have more than 1,000 occurrences and are included in the ML positive dictionary.

[36] Our previous analysis suggest a few thousand unigrams comprise most of the signals in our corpus.

bigrams (many have over 50% of their bigrams labeled as neutral), and a few show up with the opposite sentiment signs. The LM words also need significant disambiguation according to the ML algorithm.

In order to see which are the unambiguous unigrams, Table 5 presents the top/bottom 30 unigrams, out of the 500 most frequent, ranked using the difference between positive/negative ML bigrams counts (frequency weighted). These words all are associated with bigrams that are mostly positive/negative, those clustered in the left/right lower sides of Figure 3, and thus seem like natural candidates as signals of sentiment.

Turning to the positive terms, we see some obvious candidates with high frequency counts from the LM dictionaries: *pleased*, *improved*, *improvement*, *effective*, *strong*, and *great*. But 24 of the 30 words listed in the positive bigrams column do not appear in LM. It is worthwhile to note how a few of these 24 words are associated with "hard data:" balance *sheet*, *diluted share(s)*/earnings, *free cash flow*, *shares* outstanding, operating *leverage*. Human coders would be very unlikely to flag *quick* or *wondering* as positive sentiment words.

On the negative side of Table 5, we find similar conclusions. The token *issue(s)* is almost exclusively associated with negative bigrams, and so are *understand* and *impacted*, words that are unlikely to be coded as negative by humans. We note that only three of the negative words in the table (*loss*, *decline* and *negative*) are part of the LM dictionaries.

While the terms included in Table 5 clearly have some new color, it is important to note that even for these unigrams that are associated with extremely positive/negative bigrams there is still a fair amount of disambiguation. The token *not* is certainly negative, but only 49% of the time (*not really* and similar bigrams are neutral).

The token *not* deserves some further discussion, as it is standard negation in English, and it is very common in our corpus.[37] The folklore in the literature is that positive words have less impact due to such negations. Our analysis of bigrams associated with *not* can shed some light on this. There are 1,483 different bigrams in our 65K dtm that contain the token *not*. While 1,007 of these are classified as neutral, the 425 that are classified as negative by the ML algorithm comprise 45.5% of the frequencies of all bigrams that contain the token "not." But only 14 of the 425 are actually associated with LM positive words (i.e., negations of positive words), a trivial proportion of the frequency counts the ML algorithm classifies as negative (1.1% of the total frequency counts associated with "not" in our 65K dtm). We also find that only 0.1% of such "not bigrams" are classified as positive. While the ML algorithm disambiguates these

[37]The fact that a token it is very common tends to dampen its effect in our sentiment scores, as what matters is the cross-sectional variation (across earnings calls). This is true for other common tokens such as *question(s)*.

bigrams, overall it considers most negations as negative signals in the earnings calls.

As shown in Figure 3, there are many unigrams which have mixed bigrams loadings. The ML algorithm is able to disambiguate many of these, coding differently a unigram according to its company. While talking about *cash flow* is generally positive, the bigram *cash burn* is labelled as negative by the ML algorithm. Another leading example is the token *bit*, which most readers would not label as colorful. The ML model flags many of its bigrams as positive (*bit ahead/-tailwind/money/faster*), and a handful as negative (*bit softer/confused/longer/slower*).

To provide a more detailed example, consider the token *demand*. This is a particularly nuanced English word, and important in earnings calls discourse, dealing with demand for products and demands from customers/suppliers.[38] In Table 6 we list the top bigrams (by frequency) from our 65K bigram dtm that contain the token *demand*. The ML algorithm classifies roughly 1/3 of the *demand* bigrams into positive/neutral/negative categories. While none of these bigrams have large term frequencies (most of them are around 1K), they are meaningful additions to the ML sentiment scores. The tokens *increase(d)/strong demand* or *demand across* sound like positive terms, as well as *solid/healthy demand*. On the negative side we find *demand response*, *lower demand*, as well as *soft demand*, flagged as negative by the ML algorithm. It is important to note that the majority of these bigrams are not associated with words that belong to the LM list (market with $+/-$ signs in blue/red), which drives much of the improvements in the predictive exercises from Section 3.

5 10-K statements and 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. These two events, the release of the 10-K statements and the earnings calls, are obviously intimately related: the call with investors/analysts is done to discuss the financial performance of the firm, which is formally disclosed with the filing of the 10-K annual statement. This section studies 10-K statement releases, looking to see if the dictionaries constructed earnings calls have bite on this new corpus, and if ML dictionaries constructed using 10-K statements also work as well as the ones we have seen in Section 3.

[38]The token *demand* was chosen on the basis of frequency, but also lack-of-too-many-bigrams: many common words (business) have hundreds of associated bigrams, which makes it challenging to present comprehensively in a table.

5.1 10-K releases

A natural question to ask is which of the two events is more important, the earnings call or the filing of the 10-K statement. We follow Griffin (2003) and compute the absolute excess return for each day around the two events, normalized by its mean and standard deviation (computed in the period of -60 to -2 day around the earnings call date). Figure 4 plots the averages for the 10 days before and after the event for earnings calls (blue circles) and 10-K releases (red crosses). We note that the average absolute value, under normality, should be around $\sqrt{2/\pi} \approx 0.8$ (dashed line).

Figure 4 shows that earnings calls are associated with significantly more volatile stock prices than 10-K statements. The average absolute excess returns on the earnings call event date is over 2, versus 1.2 for the 10-K release event, that is earnings calls are associated with volatility that is 150% higher than on regular days, whereas for 10-K releases it is only 50% higher. This effect is still apparent on the date after the event: average absolute excess returns over 2 versus 1.[39]

The stock price reaction to the 10-K release is much more muted than the one to earnings calls. This should be not surprising, as the earnings calls often happen a week before the formal submission/acceptance of the 10-K statement by the SEC. This evidence argues that earnings calls are a better event for performing the type of supervised learning algorithm we implement in our paper. A stronger signal-to-noise ratio for the event is critical for the success of our ML algorithm. While the 10-K release is an important event for the firm, most of the information is conveyed to markets during the earnings calls that precede the 10-K release.

With this in mind, we next extend our analysis by studying the full text of the 10-K statements, as provided in Bill McDonald's webpage.[40] Our focus will be in predicting the stock market reaction over the four days around the release of the 10-K, mimicking our previous analysis and that in Loughran and McDonald (2011).

The dataset we study 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

[39] We note that when the 10-K is released on the same day or the day before the earnings call, which occurs only in 701 occasions (out of 3,809 total events) the average absolute excess return is 2.21 on the event day (1.86 the day after). For 10-K statements released one or more days after the earnings calls occurs, the majority of our sample, the average absolute normalized return is 0.93.

[40] 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.

(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 the 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 80,250 observations.

We clean/parse each of the filings as described in Section 2.1, resulting in document-term-matrices using both uni/bigram representations. We compute sentiment scores as in Section 2.3, adding the term frequencies of positive/negative words in a given 10-K statement. We note that we do not impose term frequency limits on the corpus when computing LM scores, only when training the ML algorithm (we use 65K dtms as before). Lastly, we scale all sentiment measures to unit variance so that their magnitudes can be compared across measures (ML versus LM) but also across corpora (earnings calls versus 10-K). We do the training using all 10-K releases from 1996–2004, and perform our out-of-sample exercise on the 2005–2018 data.

In Table 7, 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 LM sentiment scores are not associated with the stock price reactions during the release of 10-K statements.[41] The ML unigrams also present no predictability: the single words picked by the ML algorithm are not associated with stock returns in the out-of-sample period. The last two columns in Table 7 present the results when using bigrams. We find no predictability using the negative bigrams on the 10-K corpus, with some (weak) results for the positive bigrams (t-stats with absolute values in the range 2.9–3.3). The overall performance of the ML dictionaries is rather poor, relative to the results using the earnings calls corpus presented in Table 2.

One could have conjectured that the reason the ML dictionaries outperform the LM dictionaries in our analysis 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 very strong (see Table 2), and that their predictability in the last 15 years on 10-K statements is rather weak. It is worth emphasizing the point that the LM dictionaries perform significantly better in the earnings calls corpus, relative to the 10-K releases, which supports our interpretation of the evidence in Figure 4 regarding which events conveys more information.

[41] 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.

Our findings suggest that 10-K releases are not a good corpus where to assess the impact of soft information on stock prices. The evidence in Table 7, together with Figure 4, strongly argues that there is not that much information being conveyed to the market, relative to the information released during the earnings calls studied earlier in the paper. It is not surprising that the ML algorithm does not perform as well as with earnings calls, since the supervisor does not have a high signal-to-noise ratio. But even when working with the corpus from 10-K statements, the origin of the LM word lists, the evidence in Table 7 shows that the ML algorithm can capture some sentiment that is not measured by existing bag-of-word approaches.

5.2 External validity

In the previous analysis, we have trained our algorithm on a given corpus, and use similar textual data (earnings calls, 10-K statements) in order to test predictability out-of-sample. We have found that the ML dictionaries trained on earnings calls do extremely well out-of-sample on earnings calls, whereas training on 10-K statements yields weaker results. We now ask whether such dictionaries perform well in other corpora. This will speak to the relative merits of our ML approach relative to the corpus used to train it.

Recall that Loughran and McDonald (2011) developed their word lists adapting the standard General Inquirer (GI) dictionaries, which were developed in the psychology literature decades ago, to the lingo and nuances of financial and accounting words in the context of 10-K statements. The literature as a whole has shown such dictionaries have strong predictability in very different contexts, from newspaper articles to IPO prospectuses and a myriad of other regulatory filings (Loughran and McDonald, 2016). One of the appeals of such a "general finance jargon dictionary" is precisely its ability to use it "off-the-shelf" across different corpora. Our own analysis in Section 3 confirms the strong predictability of the LM dictionaries in the context of earnings calls.

We will compare these two dictionaries to those generated by the ML algorithm. We will use a simple empirical exercise: fit the ML algorithm to a given corpus, then use the dictionaries that come as output to the ML fit in order to measure sentiment in a new corpus. We note that there are two important parameters in this exercise: the sample period for the dictionary creation/-training, and the sample period to use for the predictability regressions (with the new corpus). We will use the dictionaries we have developed this far, using earnings calls 2006–2014 (as in Section 3), and 10-K statements from 1994–2004 (as in Section 5.1), for simplicity, being mindful of potential time overlap(s), and also statistical power of our econometric specifications.

For expositional purposes, we will not run horse race regressions among all the different dictionaries, but instead present the output from regressions where we add each of the sentiment scores separately (the results with joint estimations are virtually identical). We will report the percentage of the corpus that is covered by a given dictionary, in term frequency terms, as a measure of coverage. We also report the point estimates, which are comparable across dictionaries and corpora, since we normalize our independent variables throughout to unit variance. Finally, we report the t-statistic of the regression from that sentiment score.

We start with studying the external validity of different dictionaries using the 10-K releases as a laboratory. In Panel A of Table 8 we use the full sample of 10-K releases, 1994–2018. We find that the LM dictionaries have some bite with 10-K stock price reactions, but only the negative dictionaries, confirming some of the folklore in the literature. We note that the GI dictionaries do not have any predictive power. The last four rows of Panel A find significantly stronger predictability using the ML dictionaries constructed using earnings calls. For both unigrams and bigrams the point estimates and the absolute value of the t-stats are higher than those from either the LM or GI dictionaries.

We note how the individual coefficients/t-stats on the ML sentiment scores in Panel A of Table 8 are at least as significant as in Table 7 for positive words, and more significant for negative words. The ML dictionary trained on earnings calls performs better than both the LM/GI dictionaries, and better than the ML dictionary trained on the 10-K corpus.

Panel B in Table 8 presents the results using the earnings calls corpus from 2006–2019 as the events. The first two rows give the estimates of the LM dictionaries, which are comparable to those in Table 2: there is strong predictability using the LM dictionaries, with absolute t-stats above 8 for both positive and negative words. The next two rows give the results for the GI dictionaries, showing that they also provide significant predictability, but smaller compared to the LM dictionaries, providing new support for the dictionary refinement performed by Loughran and McDonald (2011).

The last four rows in Panel B provide the point estimates from the ML n-gram dictionaries constructed using the 10-K corpus, as in Section 5.1. We find rather weak predictability, mirroring the results from Table 7: no significance for the unigrams words, and marginal performance for the bigrams. Clearly the performance of the ML dictionaries trained on 10-K releases is significantly worse that the ML dictionaries trained on earnings calls. This should not be surprising given our previous results: the signal-to-noise ratio of the 10-K events is quite small relative to that from earnings calls, and the ML algorithm needs a strong supervisor.

In summary, the results in this section are mixed regarding the external validity of the ML

dictionaries, as constructed in Section 2.2. The n-grams selected using the earnings calls do seem to have some bite in the 10-K corpus, relative to LM, but not vice versa. This is to be expected given the pecking order among these two events. A large scale exercise trying to tease out the general external validity of different dictionaries seems like an interesting route for further research.[42] We turn now to the goal of constructing dictionaries that are further refined, attempting to get at "plain money English" that can have some general validity across time and corpora.

6 Stability and plain money English dictionaries

One of the goals of our research project is the generation of new dictionaries of words (n-grams) that can help creating sentiment scores when reading financial text. Our analysis makes clear that a large representation of the underlying text, say via a few thousand bigrams, has the best predictive power for the earnings calls corpus. But our empirical exercise can contribute by providing small unigram lists of terms that have significant color in our analysis. The construction of unigram dictionaries is the most transparent, as it maps one-to-one to the standard "bag-of-words" approach, and our previous evidence strongly suggests there are many (unigram) terms from the ML dictionaries that can contribute to sentiment measurement beyond the LM word lists. We first study the stability of our dictionary constructions, and then suggest a regularization step to define what we refer to as ML "plain money English" dictionaries.

6.1 Stability of ML classifications

As we have highlighted throughout the paper, the ML dictionary construction depends on the particular training sample chosen. In order to have a set of n-grams that have external validity, we would like to include tokens that have significant coverage of the underlying text, appear frequently, are used by many firms, and do not suffer from potential over-fitting. Ideally these new dictionaries would also be general enough that many industries use them, and they are part of business narratives throughout our sample period, not just in a small subsample. A related important question regards the stability of our dictionaries: does a given n-grams get labelled as positive/negative in all/most subsamples?

In order to address these questions, we subset our earnings call dataset into m different subsam-

[42] The corpora/events to be used is potentially large and outside the scope of our draft: from press releases (8-K), to prospectuses (IPOs), to media articles, TV interviews, FOMC minutes, etc.

ples. At this point the reader should think of these subsamples as different years or different industries. For each of these m subsamples, we estimate the MNIR model and generate ML dictionaries, following the algorithm in Section 2.2. We do the estimation using the same 65K dtm for each subsample, so for each n-gram, we have a set of m different MNIR loadings.[43] We can then ask about how many times a given n-gram has positive/negative/neutral loadings, and look for consistency across the samples.

The simplest case to consider is two subsamples, in a similar spirit to the exercise on stability considered in Section 4.2 of Jegadeesh and Wu (2013). We cut the sample into two periods, the training and OOS periods from Section 3, so that a given n-gram gets classified as positive/negative/neutral twice. Panel A in Table 9 presents the relative frequencies of each possible combination, starting with a dtm of 4,096 unigrams.[44] When training in the first-half of our sample, we get 1,065 positive unigrams, of which 564 are again estimated as positive in the second-half of our data, with 132 being (mis)classified as negative. The percentages in the table in parenthesis denote the term frequencies, in percentages, of those two groups of unigrams: the positive/positive tokens comprised 29.7% of the corpus, with only 2.8% corresponding to misclassified positive/negative words. We see a similar pattern with the negative words: out of 1,282 negative unigrams, 745 are again estimated as negative in the second-half of our sample (21.7% term-frequency), and 154 as positive (2.2% term-frequency). Overall, 60% of the term frequencies in our documents are consistently labeled positive/neutral/negative in both subsamples, with only a total of 5% that get misclassified as positive/negative.

Turning to bigrams, in Panel B we start with the 65K bigram dtm that we have used throughout the paper. Roughly 9K terms get labeled as positive/negative, with 47K labeled as neutral in both samples, as before. About 1/3 of the bigrams get put into the sample positive/negative bucket in both subsamples (2,501 positive bigrams, 2,704 negative bigrams), with a large piece of the bigrams consistently labeled as neutral (36,071). There is some misclassification, with about 700 terms getting classified as positive (negative) in one sample and negative (positive) in the other. But as the table shows, these are very infrequent instances, comprising less than 1.5% of the whole text of the earnings calls.[45] In contrast, 20.1% of the corpus is classified consistently as neutral, and another 4.4/5.5% consistently as negative/positive.

In order to push the idea of stability of the word classifications, we proceed to subset the data

[43] We note that there is the possibility that a given n-gram is not part of the corpus of one of the subsamples. We set the MNIR loading in that case to zero (neutral word).

[44] The results are identical with the 65K dtm, but as discussed in the context of Figure 1, 4,096 (2^{12}) unigrams capture more than 95% of the text in the earnings calls.

[45] Note that the coverage numbers in Panel A add up to virtually 100%, whereas those in Panel B only to 48%, since the 65K bigram dtm only covers 48% of the text of the earnings calls, see Section 4.1 and Figure 1.

across both time and industries. In particular, we consider each of the five industry groups using the Fama and French industry definitions, and subset each industry group into five different time periods. With this empirical design, we have a set of 25 different subsamples where we can fit the MNIR and construct dictionaries as before.[46] In order to get at "plain money English," we will require that: (a) the n-gram shows up with a positive/negative loading at least k times; (b) it does not show up with the opposite sign more than l times. In the calibration below, we set $k = 8$ and $l = 1$.

There are a total of 648 (844) unigrams that are positive (negative) eight or more times across the 25 cross-sections. The top panel in Figure 5 plots the number of times a given unigram is classified as positive (negative) at least eight times in blue (red), against the number of times such given n-grams are classified as negative (positive), across the 25 different subsamples. We see that there are 133 (182) unigrams that are classified as positive (negative) eight times, and all other loadings (across the other 17 cross-sections) are neutral. Similarly, there are 100 (145) such unigrams that get misclassified once. So we have a total of 233 (325) unigrams that are positive (negative) eight times or more, and get misclassified at most once. In other terms, 65% (61%) of the unigrams that are positive (negative) eight or more times get misclassified more than once.

The bottom panel in Figure 5 plots the number of times a given bigram is classified as positive (negative) at least eight times, against the number of times such n-grams are classified as negative (positive), across the 25 different subsamples. There are a total of 823 (967) bigrams that are positive (negative) more than eight times across the 25 cross-sections. We note how the bigram graph drops off much quicker than the unigram graph: there is significantly less misclassification for bigrams than for unigrams. To make things precise, out of the n-grams that are positive (negative), 352 (325) of them never get misclassified, and 216 (287) are misclassified once. So we have 568 (612) bigrams that are classified as positive (negative) at least eight times, and get misclassified at most once. In other words, 31% (37%) of the bigrams that are positive (negative) eight or more times get misclassified more than once.

To summarize, we fit the MNIR model over 25 different cross-sections, and keep only the uni/bi-grams that result in eight or more consistently positive (negative) loadings, with at most one misclassified subsample. The misclassification, using our calibration, is roughly 1/3 for bigrams versus just shy of 2/3 for unigrams. This brings up some more levers that we can use to further refine our dictionaries, which we exploit in the next section.

[46] We "cut" the time dimension to have the same number of observations along the time dimension for each industry grouping. The smallest subsample has just over 1,500 observations, with the average subsample having around 2,400 observations. We remark that if a n-gram does not appear in one of the cross-sections, we assign a neutral score to the n-gram.

6.2 Plain money English dictionaries

Our previous algorithm attempts to regularize the ML output, requiring stability along different cross-sections of the underlying data. Our method results in unigram dictionaries with 233 positive terms, and 325 negative terms. The bigram dictionaries have 568 positive terms, and 612 negative terms. These are listed in the Appendix as "plain money English dictionaries."

Note how these lists of *n*-grams are quite compact relative to our previous analysis. Despite this, the unigrams in our "plain money English dictionaries" have term frequencies of 13.5% and 10.9%, for positive and negative unigrams, and 2.2% and 1.4% for bigrams. The ML unigrams are therefore significantly larger than the standard LM dictionary (similar in term frequencies to the GI dictionaries, see Table 8), and the ML bigrams are similar in term frequencies to the LM dictionaries.

It is important to keep in mind that the cutoff numbers we use (eight and one within 25 cross-sections) are not critical, and they are fairly stringent: the lasso penalty makes many of the loadings across the 25 subsamples zero. We also note that the set of *n*-grams that are getting selected using this mechanism have several important properties: they are going to be part of at least two industries and two time periods (roughly four years), and they will need to be relatively frequent.[47] The filters that we apply are attempting to get at plain English by implicitly requiring that the tokens appear in a significant number of industry/years, and they do so in a consistent way regarding their sentiment.[48] The code and data in the depository associated with our paper provides different ways to generate such dictionaries, that the reader can customize to their needs.

Our new dictionaries are relative "thin" lists of *n*-grams: 233 positive and 325 negative unigrams. The unigram list is certainly very different than those from psychology or the standard LM dictionaries. There are several unigrams that are context specific (*driving/leverage*), but many other frequent words that convey a sense of positivity not measured in standard dictionaries (*allowed/generate/sustain*). Negative unigrams have a similar mix of terms that are clearly negative in earnings calls context (*adjustments/challenge/estimate*), but also many other terms that are less likely to be considered positive or negative by a human coder (*change/expect/factor/soft*). As expected from Table 3, the overlap between the ML unigram dictionary and the LM word lists is rather small (roughly 20% of the terms).

[47] We note that the tail end of the 65K dtm has terms with roughly a few hundred appearance in the *whole* corpus. Recall we are training with roughly 2,400 observations, while keeping constant the dtm that has over 60K terms.

[48] There is a cost to such homogeneity, i.e., we may miss some *n*-grams that are particularly positive/negative in some industry specific context.

On the bigram side, our previous calibration yields 568 positive and 612 negative bigrams. Similar qualitative comments to the unigram lists apply to bigrams. We highlight some positive terms: *able leverage/reduce/take*, *cash flow/generation*, *demand across/coming*, *improve+*, *obviously+*, *really+*, *quarter+*. And some negative terms: *competitive marketplace/pressure/pricing*, *dead horse*, *increased competition/cost/expense*, *not happen/good/hit/issue*, *senior management*, *specific issues*.

One could further try to classify the sentiment n-grams by their part-of-speech tabs (verb, noun, adjective). And we could classify regarding semantic meaning/content, from accounting terms (revenue growth), to more colloquial expressions (congrats quarter). We leave such work for future research (see Meursault, Liang, Routledge, and Scanlon, 2021, for an early effort).

We conclude this section by analyzing the sentiment scores that are generated using these new "plain money English" dictionaries. Since we want to report out-of-sample regressions, we repeat the previous steps training only on the earnings calls from 2006–2014, as in Section 3. We generate 25 different cross-sections, fit the MNIR to each of them, and apply the filters from Section 6.1, setting $k = 7$ and $l = 1$.[49] We then run a horserace as in Table 2, using the earnings calls during the period 2015–2019.

Panel A in Table 10 presents the results for earnings calls, for the period 2015–2019. In the first column of Panel A we have the results using only the ML "plain money English" words. The regularization step, requiring consistency across subsamples, does generate significantly higher predictability than that reported in Table 1: the point estimates and t-statistics are much higher, and the 6.6% R^2 is larger than any of the ones reported in Table 1 or Table 2. Adding the LM words, shown in column two, barely changes the overall fit of the regression, and the results are robust to excluding the ML words that belong to the LM dictionaries.

The results for bigrams in Panel A of Table 10 mirror those for unigrams: very strong predictability, with t-stats over 9 in absolute value, and point estimates above 1.5, with an overall fit measure of $R^2 = 6.5\%$. Adding the LM dictionaries, shown in columns 5–6, we see they have some predictive power, but most of the variation in stock returns is again being explained by the ML word lists. We emphasize that these larger improvements are obtained despite the fact that the dictionaries we are using are quite small relative to those in Tables 1–2, which used over 10,000 n-grams in each positive/negative dictionary (versus a few hundred now for both unigrams and bigrams).

[49] We use slightly softer criteria for inclusion, $k = 7$, and the same for exclusion, $l = 1$, due to the extra noise that we face when repeating the exercise with half of the data. This out-of-sample exercise, where we calibrate our model using only data from 2006–2014, yields a total of 207 (304) positive (negative) unigrams, and 538 (571) positive (negative) bigrams.

In Panel B we have the results using 10-K statements, following our analysis from Section 5.1, but using the new "plain money English" dictionaries. We find that unigrams have some bite, but very marginal statistical significance. On the other hand, bigrams have very strong predictability, with t-stats hovering around 3–4 in absolute value. We note the improvement over the LM dictionaries is significant, as the LM scores are no related to the 10-K filings stock price reactions.

To summarize, the horseraces between the LM and ML "plain money English" dictionaries yield similar results to those reported earlier in the paper. The LM word lists do have some predictive power, but both the economic magnitude and statistical significance of the ML dictionaries is higher throughout. This is true both using earnings calls, the focus of our paper, but also 10-K statements. Overall, the performance of the "plain money English" dictionaries developed in this section, comprised of a few hundred n-grams, is quite good relative to the current gold standard. Only further research, using other events/corpora, will settle the ML versus LM debate.

The data depository[50] 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).[51] 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 6.

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.

7 Conclusion

We construct dictionaries based on the machine learning algorithm of Taddy (2013), using a large corpus of earnings call transcripts. We find that the tokens chosen by our algorithm per-

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

[51] 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`.

form significantly better than the existing techniques based on bag-of-words, specially when allowing for bigrams. 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 much better than single words using 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).

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, "Managements 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.

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.

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.

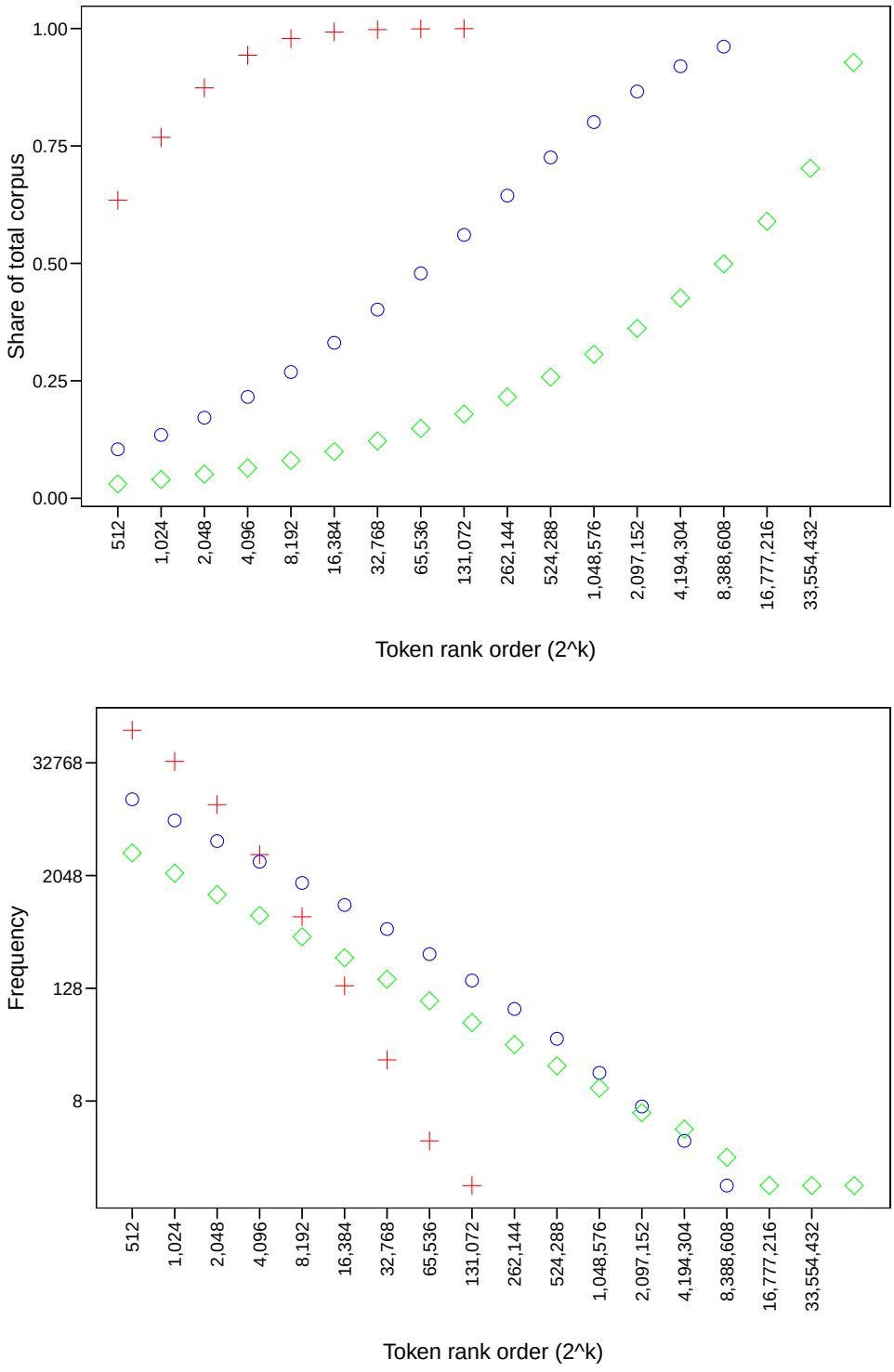
Rabinovich, M., and D. Blei, 2014, "The inverse regression topic model," in *Proceedings of the 31st International Conference on Machine Learning*, ed. by E. P. Xing, and T. Jebara, vol. 32 of *Proceedings of Machine Learning Research*, pp. 199–207, Bejing, China. PMLR.

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.

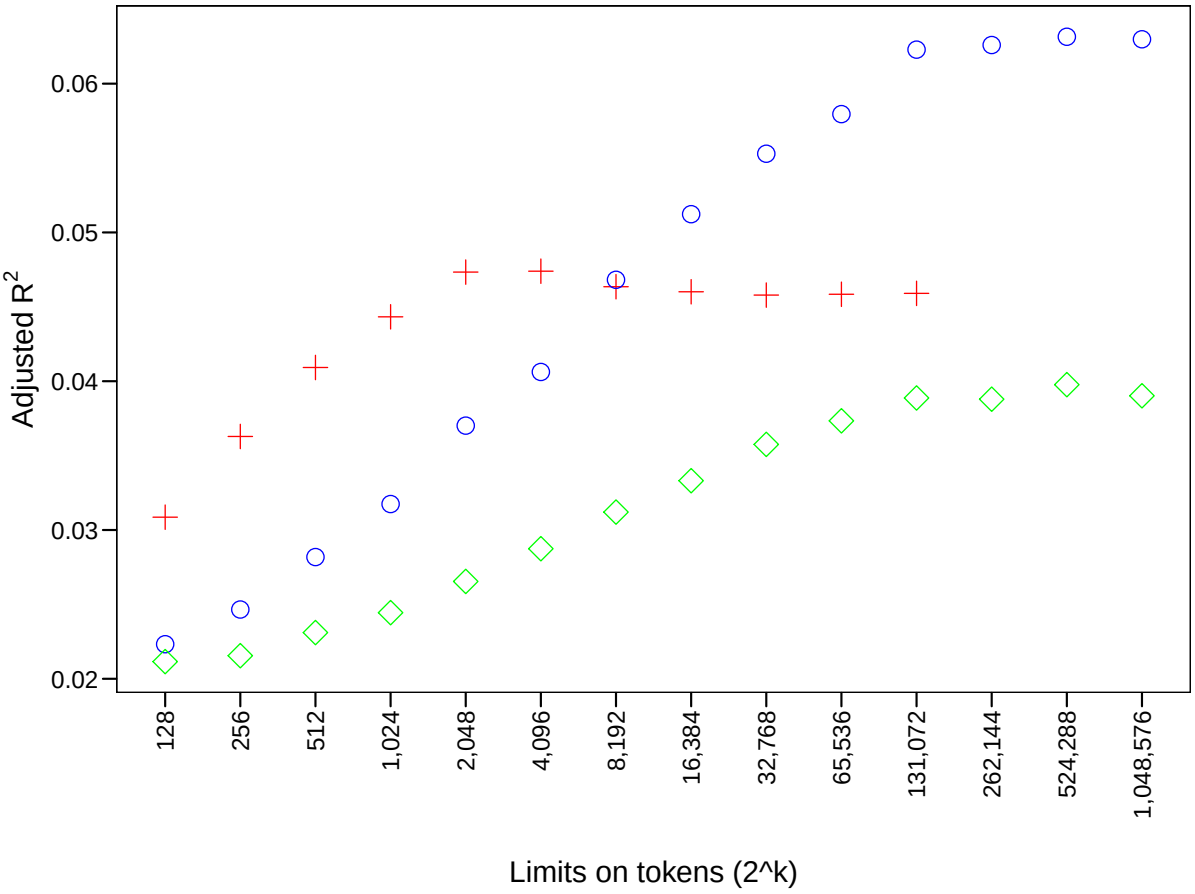
Taddy, M., 2013, "Multinomial inverse regression for text analysis," *Journal of the American Statistical Association*, 108(503), 755–770.

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



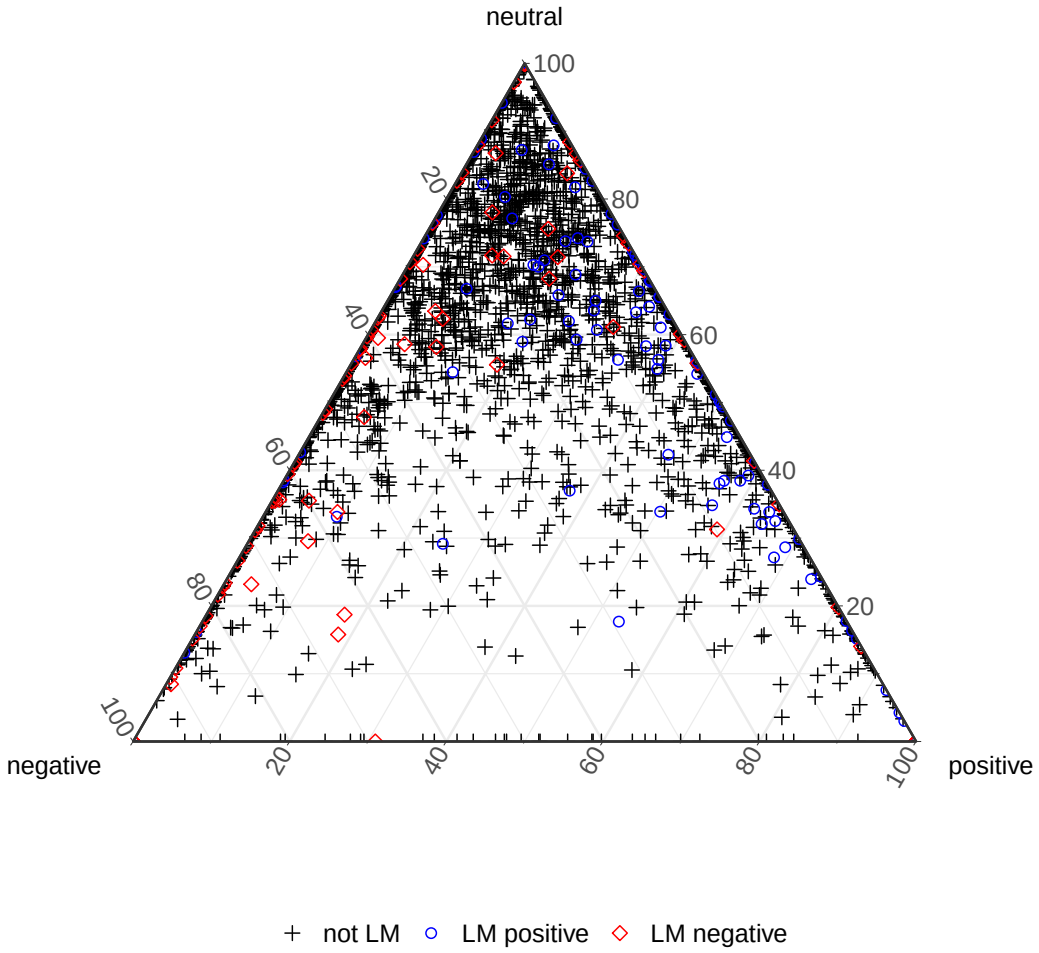
The top graph plots the proportion of the total text of earnings calls that is covered by having document-term-matrices of different sizes, starting with 512 tokens (2^9) up to 67m (2^{26}) tokens. The bottom graph plots the log-frequencies when ranking individual *n*-grams by such frequencies. The red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

Figure 1: *n*-gram coverage and log-frequencies



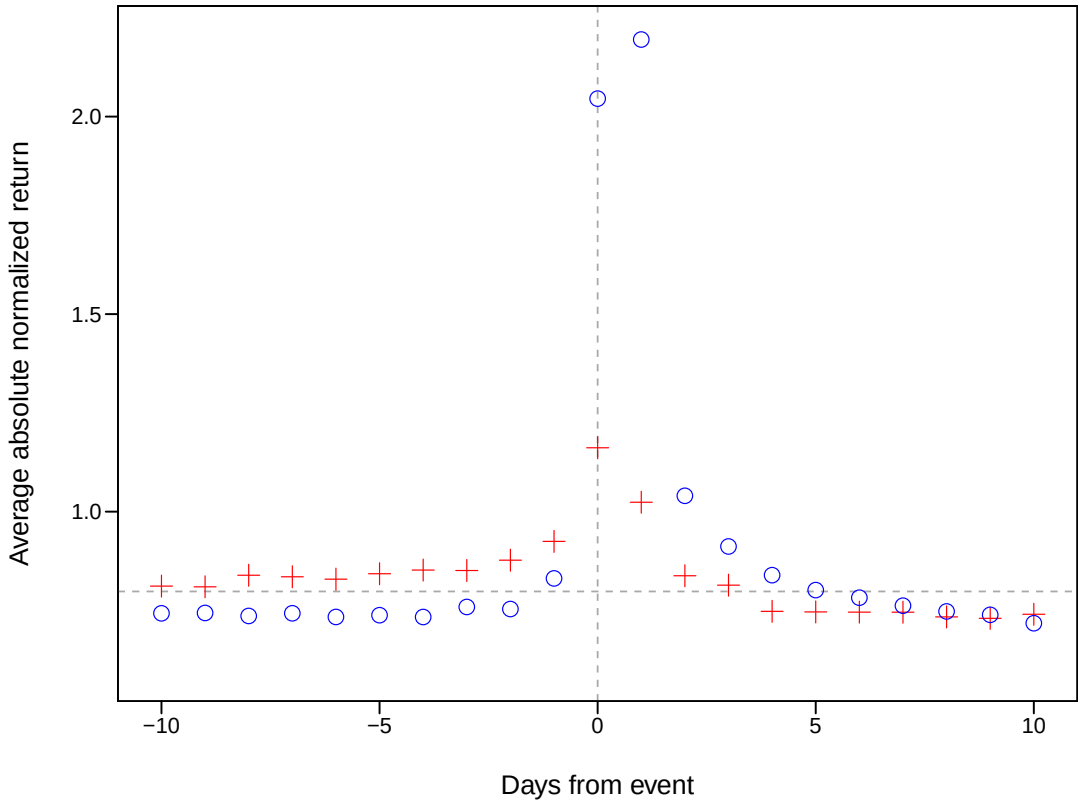
The graph plots the R^2 of a regression as in Table 1, for document-term-matrices of different sizes, starting with 128 tokens (2^7) up to 1,048,576 tokens (2^{20}). The red crosses refer to the R^2 associated with unigrams, the blue circles to bigrams, and the green diamonds to trigrams.

Figure 2: R^2 as a function of the size of the n-gram dtm



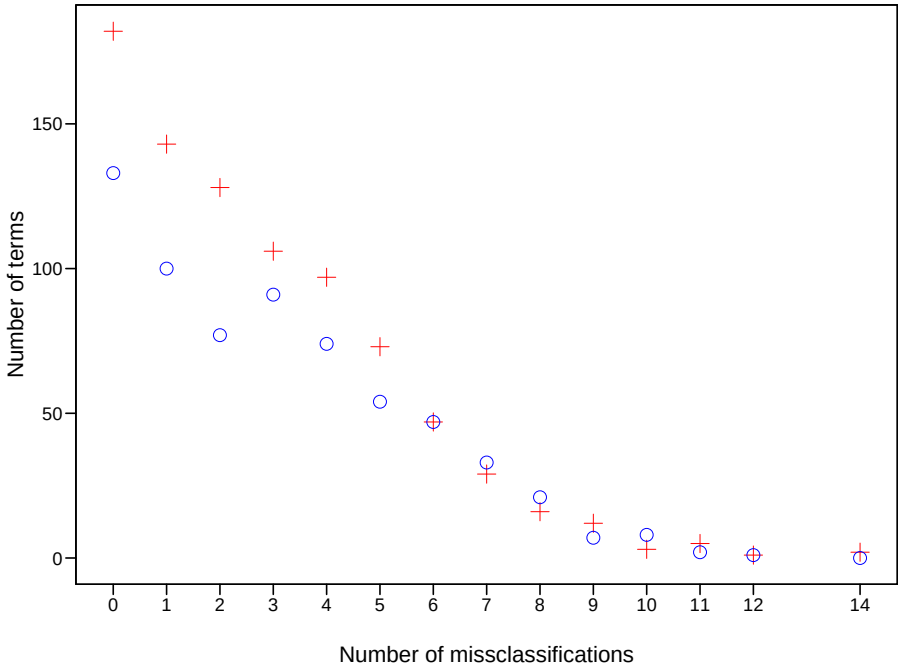
This ternary graph plots each of the unigrams that belong to a bigram dtm with 65K terms (roughly four thousand unigrams). For each unigram, we consider all bigrams that the ML algorithm classifies as positive/neutral/negative, and compute the frequency weighted fraction that are classified in each category. This results in three coordinates that add up to one, which we plot in the ternary graph. The blue circles are LM positive words, whereas the red diamonds are LM negative words. The black crosses are terms that do not belong to the LM dictionaries.

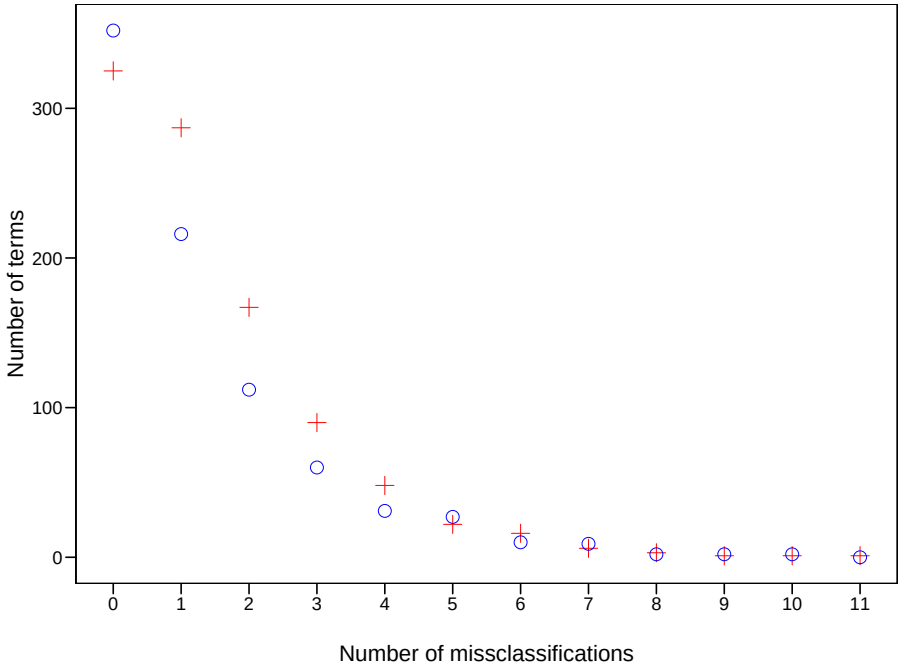
Figure 3: Unigram and bigram sentiment scores



This figure reports average absolute normalized excess return around the filing date of 10-Ks and earnings calls. The point estimates for the earnings call events are plotted with blue circles, whereas the red crosses represent the days of the filing of the 10-K report. The sample consists of 10-K releases that have a call in the window of -60 to $+1$ days around its filing. Excess return is CRSP daily stock return less the value-weighted total return index subsequently normalized by its mean and standard deviation, computed in the period of -60 to -2 days relative to the earnings call date. The dash horizontal line is the the expectation of the absolute value of a standard normal random variable.

Figure 4: Average absolute returns around filing events





We estimate the MNIR model across 25 different cross-sections, defined by five time subsets for each of the five Fama and French industry groups. We consider n-grams that are positive/negative in 8 or more of these subsamples. The top figure plots the number of such unigrams according to the number of subsamples that have the opposite sign (negative for the positive unigrams, and vice versa). The bottom figure presents the corresponding plot for bigrams. The blue circles correspond to positive terms, the red crosses to negative terms.

Figure 5: Mis-classification across 25 cross-sections

Table 1: Comparing unigrams, bigrams, and trigrams

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 Taddy's model and extract which n-grams are annotated as positive and negative. The results presented in the table correspond to earnings calls from 2015–2019. We construct the sentiment measures using term frequency weights separately for unigrams, bigrams, and trigrams. 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).

ML positive unigram	0.71^{**}			0.19
	(2.4)			(0.7)
ML negative unigram	-1.64^{***}			-0.48
	(-4.8)			(-1.4)
ML positive bigram		1.73^{***}		1.29^{***}
		(7.8)		(7.1)
ML negative bigram		-1.55^{***}		-1.38^{***}
		(-10.0)		(-8.4)
ML positive trigram			1.45^{***}	0.05
			(10.6)	(0.7)
ML negative trigram			-1.16^{***}	0.02
			(-7.1)	(0.2)
Adjusted R^2	0.046	0.058	0.037	0.059
Observations	27,644	27,644	27,644	27,644

Note: *p<0.1; **p<0.05; ***p<0.01

Table 2: Horse race regressions

This table reproduces the results in Table 1 adding the LM sentiment scores. In the first column only the LM sentiment metrics are included. In the second and third columns we consider the ML unigram scores, separating those that do not belong to the LM dictionary (No overlap). In the fourth and fifth columns we present the ML bigram scores, separating those that do not contain any LM word (No overlap). All sentiment measures are constructed using term frequency weights, and scaled to unit variance. Standard errors are clustered on FF49 industries and fiscal quarters. The table presents point estimates and t-statistics (in parenthesis).

	LM only	**ML unigrams**		**ML bigrams**	
		All tokens	No overlap	All tokens	No overlap
LM positive	1.06^{***}	0.71^{***}	0.82^{***}	0.57^{***}	0.72^{***}
	(9.6)	(5.9)	(6.0)	(5.8)	(6.9)
LM negative	-1.10^{***}	-0.73^{***}	-0.83^{***}	-0.65^{***}	-0.80^{***}
	(-6.5)	(-5.6)	(-5.9)	(-5.0)	(-5.7)
ML positive		0.64^{**}	0.69^{**}	1.39^{***}	1.16^{***}
		(2.1)	(2.6)	(6.7)	(6.1)
ML negative		-1.00^{***}	-0.77^{**}	-1.22^{***}	-1.04^{***}
		(-3.1)	(-2.5)	(-8.6)	(-8.1)
Adjusted R^2	0.045	0.055	0.053	0.064	0.059
Observations	27,644	27,644	27,644	27,644	27,644

Note: *p<0.1; **p<0.05; ***p<0.01

Table 3: Dictionary comparisons

The table provides a confusion matrix describing the overlap between the LM and the ML dictionaries. We define a uni/bigram as ML positive/negative if it has a positive/negative loading when estimated on the full set of earnings calls. We define the overlap of LM positive (negative) with ML positive (negative) as those LM words that are only members of ML positive (negative) *n*-grams (columns labelled "Positive" and "Negative"). If the LM words do not belong to any positive or negative ML *n*-gram, we refer to them as "Neutral." The "Remaining" column reports how many LM terms do not belong to the top 65K *n*-grams.

Panel A. Unigrams

	Full dictionary	ML unigrams			Remaining
		Positive	Negative	Neutral	
LM positive	354	116	44	192	2
	(3.5%)	(2.8%)	(0.2%)	(0.6%)	(0.0%)
LM negative	2,355	164	513	1,242	436
	(1.9%)	(0.2%)	(1.0%)	(0.7%)	(0.0%)

Panel B. ML bigrams and LM words

	Full dictionary	LM words			No LM word
		Positive	Negative	Both +/−	
ML positive	8,655	1,352	134	6	7,163
	(10.7%)	(1.4%)	(0.1%)	(0.0%)	(9.2%)
ML neutral	47,364	3,359	1,261	23	42,721
	(28.1%)	(1.6%)	(0.7%)	(0.0%)	(25.7%)
ML negative	9,517	357	593	6	8,561
	(9.0%)	(0.2%)	(0.4%)	(0.0%)	(8.3%)

Panel C. Breadth and coverage

Dictionary	Positive words		Negative words	
	Signals	Coverage (tf%)	Signals	Coverage (tf%)
LM	354	(3.5%)	2,164	(1.9%)
ML unigrams	8,498	(41.6%)	10,360	(30.6%)
ML bigrams	8,655	(10.7%)	9,517	(9.0%)

Table 4: LM unigrams and ML bigrams

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 corpus measured in basis points), the total number of bigrams associated with that LM term (N_b), and the percentage of positive and negative ML bigrams that contain that term (%Pos/Neg, frequency weighted).

Positive words					**Negative words**				
Token	Cov.	N_b	% Pos	% Neg	Token	Cov.	N_b	% Pos	% Neg
good	3.5	426	56.2	5.3	question	2.6	304	10.0	18.3
strong	2.6	440	62.2	3.5	questions	1.2	123	11.5	16.9
better	1.5	305	36.3	5.4	decline	0.8	179	0.8	63.5
great	1.4	185	59.0	1.8	loss	0.6	107	3.5	73.3
opportunities	1.3	234	26.6	8.4	negative	0.5	87	5.6	46.6
opportunity	1.2	229	21.2	5.1	against	0.4	53	15.2	9.2
able	1.2	238	16.0	13.7	difficult	0.4	77	1.4	42.1
positive	1.0	212	33.8	9.9	declined	0.4	67	0.4	64.1
improvement	1.0	199	68.3	4.6	restructuring	0.3	67	30.8	8.1
benefit	0.9	183	24.7	13.4	late	0.3	39	1.5	39.0
progress	0.8	132	22.1	9.1	closing	0.2	30	13.5	2.7
pleased	0.8	115	74.6	1.4	challenges	0.2	32	0.0	46.5
improved	0.7	153	69.0	2.4	losses	0.2	32	17.7	63.6
improve	0.7	147	27.0	13.7	challenging	0.2	41	5.4	36.0
best	0.7	134	21.4	12.8	closed	0.2	41	18.5	10.0
strength	0.5	93	55.9	6.1	force	0.2	16	0.6	90.9
success	0.5	76	32.6	4.1	recall	0.2	20	2.9	10.3
profitability	0.4	75	28.9	10.4	critical	0.2	23	19.0	12.7
excited	0.4	42	31.9	0.0	declines	0.2	28	0.0	42.3
effective	0.4	44	65.8	1.7	break	0.2	19	58.9	9.8
confident	0.4	42	9.4	57.4	slow	0.2	27	4.7	59.8
improving	0.4	77	38.9	4.8	volatility	0.2	17	18.7	25.7
greater	0.4	70	9.2	24.0	weakness	0.1	20	0.0	83.0
improvements	0.3	66	44.9	0.9	weak	0.1	26	6.9	29.7
gain	0.3	53	38.8	2.8	bad	0.1	16	1.9	27.8
favorable	0.3	66	58.3	3.2	challenge	0.1	13	0.0	0.0
successful	0.3	42	17.0	21.3	slower	0.1	21	0.0	60.6
stronger	0.3	56	30.3	0.0	problem	0.1	11	8.4	29.3
despite	0.3	47	16.8	13.2	negatively	0.1	17	0.0	90.5
gains	0.3	51	47.2	10.5	lost	0.1	12	0.0	84.8

Table 5: Unigrams associated with colorful bigrams

We consider the top 500 unigrams by frequency. For each of them, we compute the number of unique bigrams that contain the unigram under consideration. We then compute the percentage of the bigrams that are positive, the percentage that are neutral, and the percentage that are negative (frequency weighted). We rank all the unigrams by the difference between positive and negative bigram percentages, and present the top/bottom 30 unigrams. The columns denote, in order: the unigram (Token), its term frequency over the whole corpus (Cov., in basis points), the number of unique bigrams (N_b), and the percentage of bigrams with positive/negative loadings (%Pos/Neg). Tokens marked with a positive/negative sign (blue/red) denote LM positive/negative words.

Positive bigrams					Negative bigrams				
Token	Cov.	N_b	% Pos	% Neg	Token	Cov.	N_b	% Pos	% Neg
sheet	0.6	42	78.9	0.0	half	1.7	124	0.7	84.2
flow	1.2	123	81.5	2.8	issues	0.4	77	0.0	77.0
store	0.6	84	80.2	3.5	issue	0.4	52	0.0	72.8
diluted	0.6	37	82.6	7.7	study	0.4	35	0.0	71.5
pleased$^+$	0.8	115	74.6	1.4	loss$^-$	0.6	107	3.5	73.3
record	0.5	95	71.1	0.0	term	0.7	60	5.7	74.2
momentum	0.4	62	70.1	0.0	trying	0.7	103	2.5	67.3
free	0.5	42	75.1	6.6	seems	0.4	37	2.1	66.0
improved$^+$	0.7	153	69.0	2.4	understand	0.6	83	0.4	64.0
effective$^+$	0.4	44	65.8	1.7	impacted	0.4	81	7.5	70.9
morning	0.9	63	69.0	5.0	decline$^-$	0.8	179	0.8	63.5
improvement$^+$	1.0	199	68.3	4.6	third	2.2	247	6.5	66.3
share	2.9	372	65.2	3.4	decrease	0.4	86	3.8	60.3
points	1.3	130	64.2	2.7	month	0.5	64	5.3	57.5
fourth	2.5	261	68.9	8.5	timing	0.6	86	0.7	52.3
quick	0.4	42	63.6	3.4	second	3.6	387	7.1	58.5
leverage	0.7	113	60.5	1.5	gas	0.4	66	4.3	54.7
strong$^+$	2.6	440	62.2	3.5	associated	0.5	70	0.6	49.8
last	4.7	383	63.4	5.0	earlier	0.9	112	1.7	50.5
stores	0.8	147	61.6	3.8	long	0.7	87	2.3	50.2
great$^+$	1.4	185	59.0	1.8	care	0.4	48	10.4	57.4
shares	0.5	73	57.4	0.4	lower	1.7	335	4.3	51.2
organic	0.5	55	56.5	0.0	back	2.4	383	3.5	49.5
full	1.5	185	63.2	7.2	primarily	1.0	144	2.6	47.2
margin	2.6	363	62.8	7.1	not	10.5	1,483	1.4	45.5
wondering	0.6	62	58.1	2.7	economic	0.4	99	2.8	45.9
per	2.1	218	64.1	10.5	first	5.3	648	5.6	47.9
real	0.6	75	55.1	2.5	make	1.7	259	3.6	45.3
cash	2.7	367	61.7	9.2	negative$^-$	0.5	87	5.6	46.6
model	0.7	81	58.0	5.6	sure	1.2	102	5.1	44.9

Table 6: Disambiguating the token "demand"

This table presents a subset of the bigrams associated with the token "demand," a total of 228 unique bigrams (using our dtm with 2^{16} terms). The column "Term" lists the bigram. The column "Freq in %" is the relative counts of the bigram out of all the bigrams that contain "demand," i.e. 5.1% of the times "demand" is written, it is within the bigram "strong demand". Tokens marked with a positive/negative sign (blue/red) denote LM positive/negative words.

Positive bigrams		**Neutral bigrams**		**Negative bigrams**	
Term	Freq in %	Term	Freq in %	Term	Freq in %
strong^{+}demand	5.1	market demand	2.6	supply demand	1.8
demand products	2.1	customer demand	2.5	lower demand	1.0
increased demand	2.0	demand environment	1.8	demand response	1.0
growing demand	1.0	consumer demand	1.5	demand side	0.8
see demand	1.0	meet demand	1.1	demand market	0.6
demand across	0.9	demand new	0.9	weak^{-}demand	0.5
increase demand	0.9	demand growth	0.9	demand well	0.4
good^{+}demand	0.9	overall demand	0.9	reduced demand	0.4
up demand	0.7	more demand	0.9	weaker^{-}demand	0.4
demand services	0.7	global demand	0.9	current demand	0.4
demand seeing	0.6	increasing demand	0.8	demand generation	0.3
solid demand	0.6	demand trends	0.8	demand drivers	0.3
loan demand	0.5	high demand	0.8	seasonal demand	0.3
quarter demand	0.5	demand product	0.8	demand supply	0.3
demand coming	0.5	demand not	0.7	soft demand	0.3
stronger^{+}demand	0.4	higher demand	0.7	demand over	0.3
robust demand	0.4	underlying demand	0.7	demand pricing	0.2
driving demand	0.4	demand strong^{+}	0.7	softer demand	0.2
level demand	0.4	demand continues	0.6	decline^{-}demand	0.2
retail demand	0.4	demand customers	0.6	real demand	0.2
healthy demand	0.4	think demand	0.6	expected demand	0.2
demand think	0.3	terms demand	0.6	not demand	0.2
demand solutions	0.3	lot demand	0.6	changes demand	0.2
demand increase	0.2	demand going	0.6	demand outlook	0.2
strength^{+}demand	0.2	seeing demand	0.6	demand marketplace	0.2
demand profile	0.2	growth demand	0.6	peak demand	0.2
demand within	0.2	demand remains	0.6	demand due	0.2
saw demand	0.2	expect demand	0.5	softening demand	0.2
pickup demand	0.2	drive demand	0.5	user demand	0.2
demand saw	0.2	continued demand	0.5	demand demand	0.2

Table 7: Horse race regressions using 10-K statements

This table reproduces the results in Table 2 using 10-K release dates as the event under consideration. The ML dictionaries are constructed using the 10-K corpus for the period 1994–2004, whereas the out-of-sample regressions in the table consider the period 2005–2018. In the first column only the LM sentiment metrics are included. In the second and third columns we consider the ML unigram scores, separating those that do not belong to the LM dictionary (No overlap). In the fourth and fifth columns we present the ML bigram scores, separating those that do not contain any LM word (No overlap). All sentiment measures are constructed using term frequency weights, and scaled to unit variance. Standard errors are clustered on FF49 industries and fiscal quarters. The table presents point estimates and t-statistics (in parenthesis).

	LM only	**ML unigrams**		**ML bigrams**	
		All tokens	No overlap	All tokens	No overlap
LM positive	−0.02	0.02	0.01	0.01	0.02
	(−0.6)	(0.3)	(0.2)	(0.1)	(0.3)
LM negative	0.02	0.04	0.02	0.06	0.08
	(0.5)	(0.9)	(0.3)	(1.2)	(1.6)
ML positive		−0.02	−0.15	0.41***	0.43***
		(0.0)	(−0.4)	(2.9)	(3.3)
ML negative		−0.23	−0.36	0.16	0.17
		(−0.7)	(−1.2)	(1.4)	(1.6)
Adjusted R^2	0.005	0.005	0.005	0.006	0.006
Observations	41,724	41,724	41,724	41,724	41,724

Note: *p<0.1; **p<0.05; ***p<0.01

Table 8: External validity: dictionaries across different corpora

The table considers stock return 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 of our measures of sentiment, and X_{jt} are controls. In Panel A we consider a 4-day window around the release of a 10-K statement, and in Panel B we consider a 4-day window around earnings calls. Rows labelled LM/GI refer to sentiment scores computed using the LM or GI dictionaries. In Panel A, the ML dictionaries (uni/bigram) are constructed using the earnings call corpus, the dictionaries used in Tables 1–2. In Panel B, the ML dictionaries (uni/bigram) are constructed using the 10-K corpus, as in Table 7. Regression include one textual measure (the corresponding row) and controls. The table presents the coverage, in terms of frequency counts for each dictionary (Cov., in percentage points), its associated sentiment score point estimates (Coef.), and its t-statistics (t-stat). All sentiment measures are scaled to unit variance. Standard errors are clustered on FF49 industries and fiscal quarters.

	Cov.	Coef.	t-stat
Panel A. Predicting 10-K release stock returns (1994-2018)			
M positive	1.3	−0.10	−1.6
LM negative	3.2	−0.11	−2.2
GI positive	10.0	−0.10	−1.2
GI negative	4.2	0.00	0.0
ML positive unigrams earnings calls	26.7	0.25	4.7
ML negative unigrams earnings calls	37.6	−0.52	−5.3
ML positive bigrams earnings calls	5.0	0.13	4.6
ML negative bigrams earnings calls	7.4	−0.15	−2.4
Panel B. Predicting earnings calls stock returns (2006-2019)			
LM positive	3.5	1.17	13.5
LM negative	1.9	−1.26	−8.5
GI positive	12.0	0.60	12.6
GI negative	5.1	−0.65	−7.4
ML positive unigrams 10-K	39.2	−0.03	−0.5
ML negative unigrams 10-K	43.1	−0.16	−2.4
ML positive bigrams 10-K	7.4	0.19	3.7
ML negative bigrams 10-K	6.2	−0.15	−2.3

Table 9: Stability of ML dictionaries

The table considers both unigrams and bigrams, classifying them as positive/negative/neutral in two subsamples, split along the time dimension. The "Early subperiod" consists of all earnings calls prior to January 1st 2015, and the "Late subperiod" consists of the rest. The table presents the number of *n*-grams that get classified as positive/negative/neutral, as well as their associated term frequencies across the whole corpus (in parenthesis). Panel A presents the results for unigrams, starting with the top 4,096 by frequency, Panel B replicates the analysis for bigrams, including the top 65K by frequency.

Panel A. Unigrams

	Late sub-period			
Early sub-period	Positive	Neutral	Negative	Total
Positive	564	346	132	1,065
	(29.7%)	(6.2%)	(2.8%)	
Neutral	347	897	405	1,749
	(6.7%)	(10.5%)	(7.5%)	
Negative	154	506	745	1,282
	(2.2%)	(7.0%)	(21.7%)	
Total	1,042	1,649	1,405	

Panel B. Bigrams

	Late sub-period			
Early sub-period	Positive	Neutral	Negative	Total
Positive	2,501	5,445	694	8,525
	(5.5%)	(4.3%)	(0.8%)	
Neutral	5,337	36,071	6,027	47,586
	(4.0%)	(20.1%)	(4.2%)	
Negative	687	6,070	2,704	9,425
	(0.6%)	(4.0%)	(4.4%)	
Total	8,640	47,435	9,461	

Table 10: Plain money English horserace regressions

The following table presents horserace regressions as in Table 2 where we use the "plain money English" dictionaries introduced in Section 6, when trained over the earnings calls 2006–2014. We present results for the 2015–2019 sample of earnings calls (Panel A) and the full sample of 10-K statement releases (Panel B). All sentiment measures are constructed using term frequency weights, and scaled to unit variance. Standard errors are clustered on FF49 industries and fiscal quarters. The table presents point estimates and t-statistics (in parenthesis).

Panel A. Earnings calls

	ML unigrams			**ML bigrams**		
	All tokens		No overlap	All tokens		No overlap
LM positive		0.32^{***}	0.58^{***}		0.60^{***}	0.78^{***}
		(3.0)	(5.4)		(6.2)	(7.6)
LM negative		-0.29^{**}	-0.58^{***}		-0.48^{***}	-0.73^{***}
		(-2.4)	(-4.2)		(-4.5)	(-5.4)
Plain ML positive	1.20^{***}	1.01^{***}	0.83^{***}	1.52^{***}	1.28^{***}	0.98^{***}
	(7.4)	(6.5)	(5.9)	(11.8)	(9.9)	(8.6)
Plain ML negative	-1.58^{***}	-1.41^{***}	-1.11^{***}	-1.58^{***}	-1.31^{***}	-1.08^{***}
	(-9.6)	(-9.4)	(-8.4)	(-9.2)	(-8.8)	(-7.8)
Adjusted R^2	0.066	0.068	0.062	0.065	0.070	0.062
Observations	27,644	27,644	27,644	27,644	27,644	27,644

Panel B. 10-K statement releases

	ML unigrams			**ML bigrams**		
	All tokens		No overlap	All tokens		No overlap
LM positive		-0.11^{*}	-0.11^{*}		-0.10^{*}	-0.11^{*}
		(-1.8)	(-1.7)		(-1.7)	(-1.7)
LM negative		-0.08	-0.10		-0.02	-0.04
		(-1.2)	(-1.5)		(-0.4)	(-1.0)
Plain ML positive	0.07^{*}	0.06	0.05	0.27^{***}	0.27^{***}	0.26^{***}
	(2.0)	(1.5)	(1.0)	(4.0)	(3.8)	(3.7)
Plain ML negative	-0.15^{**}	-0.08	0.00	-0.24^{***}	-0.23^{***}	-0.17^{***}
	(-2.1)	(-1.1)	(0.0)	(-2.8)	(-2.9)	(-2.7)
Adjusted R^2	0.012	0.012	0.012	0.013	0.013	0.013
Observations	80,250	80,250	80,250	80,250	80,250	80,250

Note: *p<0.1; **p<0.05; ***p<0.01

Appendix

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 (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[52] 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.

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

Other analysis

Table 11 repeats Table 1 including all controls, as well as a specification without any text variables. It is worthwhile noticing the R^2 of 1.9% in the first column of Table 11, which is not reported in Table 1, but discussed in the main body of the paper.

In Table 12 we present our out-of-sample estimates when parsing the earnings call into its two parts: its introduction (scripted, read by one or more of the senior managers), and its questions and answers section (which includes remarks made by investors/analysts on top of managers' answers).

We proceed as before, training the MNIR model using the earnings calls prior to 2015, generating two separate ML sentiment dictionaries, one associated with the "Intro" of the call, another with the "Q&A" section. Armed with these dictionaries, we can then compare their performance out-of-sample, using the earnings calls in 2015–2019. We emphasize that the exercise is identical to the previous analysis, now with different pieces of the earnings call.

In Panel A, we train the MNIR model on the Introduction of all earnings calls 2006–2014. We use these dictionaries to compute sentiment scores in the earnings calls 2015–2019, where in the first two columns we use only the Introduction of the calls, in columns 3–4 the Q&A section, and the whole earnings call in the last two columns. Panel B repeats the exercise, training the MNIR model on the Q&A section of the earnings calls 2006–2014.

We first note that the dictionaries developed using the Introduction perform better than those developed using the Q&A section when computing scores using only the Introduction, as expected. Similarly, the dictionaries developed using the Q&A section perform better than those developed using the Introduction when computing scores using only the Q&A section. As a consequence, both dictionaries perform equally well on the whole earnings calls corpus, with R^2 around 4% for unigrams, 5% for bigrams. But we note that training on the whole earnings call corpus yielded stronger results (Table 1). It is interesting to see how unigrams perform relatively better when trained on the Q&A section, relative to training on the Introduction. On the other hand, bigrams perform equally well across both corpora.

Table 13 repeats the exercise in Table 1, but the sentiment scores are computed multiplying the term frequencies times the MNIR Z score. We see that keeping the "size" of the sentiment, as estimated by the MNIR model, does improve the OOS fit, but the improvements are relatively modest.

Table 14 repeats the exercise in Table 1 using unigrams, computing the sentiment scores separately for five different groups of words, classified according to their frequency over the whole corpus. All frequency slices seem to bring something to the table, with slightly weaker coefficients for the most frequent (Q1) and the least frequent (Q5). Table 15 does the same estimation using bigrams. We find that the most frequent bigrams have little explanatory power relative to the other four groups (Q2–Q5), with most of the predictability stemming from the middle three quantiles (Q2–Q4).

Table 11: Comparing unigrams, bigrams, and trigrams (with full controls)

This table reproduces Table 1 including all controls.

Dependent variable: Event period excess return					
ML positive unigram		0.71^{**}			0.19
		(2.4)			(0.7)
ML negative unigram		-1.64^{***}			−0.48
		(−4.8)			(−1.4)
ML positive bigram			1.73^{***}		1.29^{***}
			(7.8)		(7.1)
ML negative bigram			-1.55^{***}		-1.38^{***}
			(−10.0)		(−8.4)
ML positive trigram				1.45^{***}	0.05
				(10.6)	(0.7)
ML negative trigram				-1.16^{***}	0.02
				(−7.1)	(0.2)
log(Size)	−0.02	-0.17^{***}	-0.28^{***}	-0.21^{***}	-0.28^{***}
	(−0.4)	(−2.9)	(−4.3)	(−3.3)	(−4.0)
log(Book-to-market)	0.09	0.22	0.27^{**}	0.14	0.28^{**}
	(0.7)	(1.6)	(2.2)	(1.2)	(2.1)
log(Share turnover)	−0.06	0.17^{*}	0.16	0.06	0.17^{*}
	(−0.6)	(1.7)	(1.6)	(0.6)	(1.7)
SUE	1.09^{***}	0.96^{***}	0.91^{***}	0.98^{***}	0.91^{***}
	(5.9)	(6.3)	(6.5)	(6.0)	(6.5)
Pre FFAlpha	-2.40^{**}	-4.02^{***}	-4.68^{***}	-3.49^{***}	-4.74^{***}
	(−2.4)	(−3.4)	(−4.1)	(−3.3)	(−4.1)
NASDAQ dummy	−0.03	0.15	0.14	0.08	0.14
	(−0.3)	(1.1)	(1.1)	(0.6)	(1.1)
Fiscal quarter-year fixed effects	yes	yes	yes	yes	yes
Industry fixed effects	yes	yes	yes	yes	yes
Adjusted R^2	0.019	0.046	0.058	0.037	0.059
Observations	27,644	27,644	27,644	27,644	27,644

Note: *p<0.1; **p<0.05; ***p<0.01

Table 12: Introduction versus Q&A sections

The following table presents regressions as in Table 1, where we decompose the earnings calls into their "Introduction" and "Q&A" sections. In Panel A we fit the MNIR model to only the Introduction of the earnings calls in the training sample 2006–2014. With the dictionaries generated from this step, we reproduce Table 1 using the 2015–2019 sample, breaking down the estimation with regards to only the Introduction (first two columns), the Q&A section of the call (columns 3 and 4), or the full earnings calls (last two columns). In Panel B we fit the MNIR model to only the Q&A section of the earnings calls in the training sample 2006–2014. With the dictionaries generated from this step, we repeat the exercise in Panel A using the 2015–2019 sample, All sentiment measures are constructed using term frequency weights, and scaled to unit variance. Standard errors are clustered on FF49 industries and fiscal quarters. The table presents point estimates and t-statistics (in parenthesis).

Panel A. Model trained on Introduction

Corpus →	**Introduction**		**Q&A**		**Full call**	
ML positive unigram	-0.04		0.26^{**}		-0.45^{**}	
	(-0.2)		(2.1)		(-2.2)	
ML negative unigram	-1.90^{***}		-1.33^{***}		-2.35^{***}	
	(-4.6)		(-5.6)		(-5.2)	
ML positive bigram		1.35^{***}		1.14^{***}		1.44^{***}
		(8.8)		(10.7)		(9.6)
ML negative bigram		-1.16^{***}		-1.03^{***}		-1.45^{***}
		(-8.3)		(-8.7)		(-9.4)
Adjusted R^2	0.039	0.042	0.034	0.034	0.041	0.047
Observations	27,565	27,565	27,029	27,029	27,644	27,644

Panel B. Model trained on Q&A

Corpus →	**Introduction**		**Q&A**		**Full call**	
ML positive unigram	0.42^{***}		0.50^{**}		0.61^{**}	
	(2.7)		(2.4)		(2.5)	
ML negative unigram	-0.58^{***}		-1.40^{***}		-1.21^{***}	
	(-6.6)		(-7.6)		(-8.8)	
ML positive bigram		1.12^{***}		1.48^{***}		1.83^{***}
		(8.9)		(8.9)		(8.5)
ML negative bigram		-0.41^{***}		-1.23^{***}		-1.08^{***}
		(-4.7)		(-11.0)		(-9.2)
Adjusted R^2	0.026	0.029	0.043	0.052	0.039	0.050
Observations	27,565	27,565	27,029	27,029	27,644	27,644

Note: *p<0.1; **p<0.05; ***p<0.01

Table 13: Using MNIR weights

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. For earnings calls prior to 2015, we train Taddy's model and extract which n-grams are annotated as positive and negative. The results presented in the table correspond to earnings calls from 2015–2019. We construct the sentiment measures using term frequency weights multiplied by their associated Z score from the MNIR model. 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).

ML positive unigram	1.49^{***}			0.21^{*}
	(10.7)			(1.8)
ML negative unigram	1.17^{*}			-0.48
	(1.8)			(-1.0)
ML positive bigram		1.78^{***}		1.80^{***}
		(8.3)		(9.2)
ML negative bigram		1.63^{***}		1.67^{***}
		(6.0)		(7.4)
ML positive trigram			1.60^{***}	0.11
			(13.3)	(1.2)
ML negative trigram			1.21^{***}	0.03
			(7.7)	(0.3)
Adjusted R^2	0.041	0.071	0.045	0.072
Observations	27,644	27,644	27,644	27,644

Note: *p<0.1; **p<0.05; ***p<0.01

Table 14: Sentiment across the frequency spectrum — unigrams

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. For earnings calls prior to 2015, we train Taddy's model and extract which n-grams are annotated as positive and negative. The results presented in the table correspond to earnings calls from 2015–2019. We construct the sentiment measures using term frequency weights, separately for five groups of words, according to their frequency. Starting with a dtm of 65K terms, ranked by term frequency across the whole corpus, we take the first 41 unigrams (by frequency) and put them in the group Q1, the following 115 unigrams into Q2, the following 273 into Q3, the following 795 into Q4, and the rest into Q5. The cutoffs are chosen so that each group of words has the same term frequency over the whole corpus, roughly 20% each. 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).

Q1 Positive	0.91***					0.26**
	(7.8)					(2.2)
Q1 Negative	−0.60***					−0.17**
	(−7.5)					(−2.0)
Q2 Positive		0.99***				0.54***
		(8.9)				(4.8)
Q2 Negative		−0.84***				−0.41***
		(−12.7)				(−3.5)
Q3 Positive			0.90***			0.27**
			(7.3)			(2.1)
Q3 Negative			−1.01***			−0.53***
			(−6.5)			(−4.6)
Q4 Positive				0.87***		0.53***
				(10.8)		(5.4)
Q4 Negative				−1.34***		−0.59**
				(−3.1)		(−2.2)
Q5 Positive					0.73***	0.46***
					(6.0)	(4.0)
Q5 Negative					−1.37***	−0.19
					(−2.7)	(−0.6)
Adjusted R^2	0.026	0.031	0.036	0.039	0.033	0.048
Observations	27,644	27,644	27,644	27,644	27,644	27,644

Note: *p<0.1; **p<0.05; ***p<0.01

Table 15: Sentiment across the frequency spectrum — bigrams

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. For earnings calls prior to 2015, we train Taddy's model and extract which n-grams are annotated as positive and negative. The results presented in the table correspond to earnings calls from 2015–2019. We construct the sentiment measures using term frequency weights, separately for five groups of words, according to their frequency. Starting with a dtm of 65K terms, ranked by term frequency across the whole corpus, we take the first 410 bigrams (by frequency) and put them in the group Q1, the following 2,416 bigrams into Q2, the following 7,260 into Q3, the following 17,061 into Q4, and the rest into Q5. The cutoffs are chosen so that each group of words has the same term frequency over the whole corpus, roughly 10% each (the 65K dtm has coverage of 48%). 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).

Q1 Positive	0.91^{***}					-0.12
	(7.7)					(-0.7)
Q1 Negative	-0.57^{***}					-0.12
	(-7.3)					(-1.4)
Q2 Positive		1.55^{***}				0.61^{***}
		(10.8)				(5.4)
Q2 Negative		-1.08^{***}				-0.40^{***}
		(-9.0)				(-4.1)
Q3 Positive			1.55^{***}			0.78^{***}
			(11.8)			(8.2)
Q3 Negative			-1.20^{***}			-0.70^{***}
			(-6.5)			(-4.1)
Q4 Positive				1.36^{***}		0.51^{***}
				(7.7)		(3.6)
Q4 Negative				-1.03^{***}		-0.50^{***}
				(-7.8)		(-5.3)
Q5 Positive					1.02^{***}	0.36^{***}
					(8.0)	(3.4)
Q5 Negative					-1.08^{***}	-0.49^{***}
					(-9.9)	(-7.8)
Adjusted R^2	0.026	0.047	0.054	0.047	0.041	0.069
Observations	27,644	27,644	27,644	27,644	27,644	27,644

Note: *p<0.1; **p<0.05; ***p<0.01

ML "plain money English" dictionaries

We label with + and − signs (red and blue) all unigrams that belong to the LM positive and negative word lists. We label with + and − signs (red and blue) all bigrams that contain one word the belong to LM positive and negative word lists. We label with both signs (orange) the one bigram that contains both a positive and negative LM word.

There are 46 positive LM words and 1 negative LM word in the ML positive unigram dictionaries (out of a total of 233), and 5 positive LM words and 55 negative LM words in the ML negative unigram dictionaries (out of a total of 325).

The 568 ML positive bigrams contain 209 positive LM words, and 3 negative LM words. The 612 ML negative bigrams contain 100 negative LM words, and 23 positive LM words.

Positive unigrams: ability, above, acceleration, achieved^{+}, achieving^{+}, across, add, additive, adjusted, advantage^{+}, ago, ahead, allowed, also, amazing, anything, area, available, beat, becoming, benefit^{+}, benefited^{+}, benefiting^{+}, benefits, better^{+}, breadth, broad, buyback, buybacks, categories, cetera, color, congrats, congratulations, consecutive, consistently, continue, continued, continues, continuing, continuous, contributed, contributing, contributor, count, curious, dedication, deliver, delivered, delivering, demonstrates, demonstrating, diluted, discipline, disciplined, diverse, dividends, doubling, driver, drivers, driving, drove, effective^{+}, efficiencies^{+}, efficiently^{+}, enabled^{+}, enabling^{+}, encouraging^{+}, exceed, exceeded, exceeding, excellence^{+}, excellent^{+}, exceptional^{+}, exceptionally^{+}, excited^{+}, exciting^{+}, executing, expanded, expanding, expansion, family, fantastic^{+}, far, favorable^{+}, favorably^{+}, fifth, finding, floating, flow, flows, focused, footprint, fourth, fueled, fun, gain^{+}, gaining^{+}, gains^{+}, generated, generating, geographic, globe, goal, good^{+}, great^{+}, grew, grow, growing, grown, growth, guys, happy^{+}, healthy, helped, helpful, helping, helps, highest^{+}, highlight, highlights, hopefully, huge, importance, impressive^{+}, improved^{+}, improvement^{+}, improvements^{+}, improving^{+}, income, increase, increased, increasing, inflection, innovate^{+}, innovative^{+}, interesting, job, keeping, leverage, leveraged, leveraging, life, lot, love, luxury, margin, maybe, might, momentum, nearly, nice, nicely, operating, operations, opportunistic^{+}, opportunity^{+}, organic, outperformance, outperformed^{+}, outstanding, over, paying, penetration, percent, percentages, performance, phenomenal, pleased^{+}, points, positioned, posted, pretty, progress^{+}, proud, pumps, raising, ratio, real, really, record, records, regions, represent, representing, represents, repurchase, repurchased, results, robust, row, runway, see, seeing, segments, serving, shareholder, shares, sheet, solid, spectrum, starting, stay, steady, strength^{+}, strengthening^{+}, strong^{+}, stronger^{+}, strongest^{+}, success^{+}, sustain, sustainability, sustainable, tailwind, talk, teams, terms, terrific, testament, thank, thanks, thoughts, thrilled, ticket, traction, update, uptick, verticals, well, wondering, world, worldwide, years.

Negative unigrams: absorption, action, actions, additional, address, addressed, addressing, adjust, adjusting, adjustment, adjustments, advance, adverse^{+}, adversely^{+}, advisors, affected, affecting, affects, aggressive, aggressively, albeit, already, although, anticipated, assess, associated, assumptions, attributable, back, based, basically, became, begin, behind, believe, below, breakeven, bright, bucket, caused, causing, certain, challenge^{+}, challenged^{+}, challenges^{+}, challenging^{+}, change, changed, changes, changing, clarify, clear, closure^{+}, competition, concern^{+}, condition, confidence, confident^{+}, consequently, context, contracted, contraction^{+}, contractual, correct, cost, costs, current, cut^{+}, cycles, dealing, deceleration, decided, decision, decline^{+}, declined^{+}, declines^{+}, declining^{+}, decrease, decreased, decreases, delay^{+}, delayed^{+}, delays^{+}, delta, deployments, described, despite^{+}, deterioration^{+}, difference, difficult^{+}, difficulty^{+}, disappointed^{+}, disappointing^{+}, discussed, disruption^{+}, down, drop, due, dynamics, earlier, effects, erosion^{+}, estimate, estimated, evaluating, excuse, expectation, expected, expecting, expenses, experienced, experiencing, explain, face, faced, facing, factor, factored, factors, fell, felt, filed, fix, flat, forecast, get, given, guess, happen, happened, hired, historical, hit, hoped, hospitals, however, hurt^{+}, impact, impacted, impacting, impacts, implementation, including, incur, incurred, information, informed, initially, intact, intended, internal, issue, issues, items, knew, known, lack^{+}, large, largely, late^{+}, later, lighter, likely, limited, literally, longer, losing^{+}, loss^{+}, losses^{+}, lost^{+}, lower, lowered, lowering, lung, magnitude, make, materialize, mean, meet, miss^{+}, missed^{+}, mitigate, month, months, nature, navigate, near, necessary, needed, negative^{+}, negatively^{+}, normal, normally, not, occur, occurred, oems, offset, okay, opening, originally, output, path, pause, pending, period, pieces, place, poor^{+}, portion, pressure, pressures, primarily, principally, process, pronounced, prospects, protect, prudent, push, pushed, queue, quickly, ramp, ran, react, reality, reasons, rebound^{+}, receiving, recognize, reconcile, recover, reduce, reduced, reducing, related, remain, remains, replaced, require, required, requires, reset, residual, resolve^{+}, resolved, resources, review, revised, rigs, roles, rooms, salary, satisfied^{+}, saying, seek, senior, settle, severe^{+}, shifted, shifting, shifts, short, shortfall^{+}, simply, situation, situations, slow^{+}, slowdown^{+}, slowed^{+}, slower^{+}, slowing^{+}, soft, softening, softer, softness, somebody, sorry, specific, spending, spot, step, steps, storm, struggling, subsequent, supposed, surprise, take, taken, taking, talking, temporary, term, therefore, though, thought, time, timing, took, tough, transactional, transition, triple, trying, unable^{+}, uncertainty, under, underperforming^{+}, understand, unexpected^{+}, unfavorable^{+}, unfortunately^{+}, unusually, volatility^{+}, waiting, weak^{+}, weakening^{+}, weaker^{+}, weakness^{+}, went, winter, worse^{+}, wrong^{+}.

Positive bigrams: ability leverage, able leverage$^+$, able reduce$^+$, able take$^+$, above expectations, above guidance, above high, above top, accelerating growth, achieved record$^+$, across board, across business, add congratulations, addition strong$^+$, adjusted margin, again pleased$^+$, ahead expectations, ahead guidance, also benefited$^+$, also good$^+$, also raising, also strong$^+$, also up, anything unusual, approved new, approximately shares, around globe, around world, backlog strong$^+$, based strong$^+$, basis point, beginning see, better anticipated$^+$, better execution$^+$, better expected$^+$, better mix$^+$, better weather$^+$, bit faster, board authorized, building momentum, business up, came better$^+$, capacity utilization, capital management, cash flow, cash generating, cash generation, certainly helped, clearly demonstrates, clearly strong$^+$, combined continued, coming fruition, companies really, comps up, congrats good$^+$, congrats great$^+$, congrats quarter, congratulations again, congratulations good$^+$, congratulations great$^+$, congratulations nice, congratulations quarter, congratulations strong$^+$, congratulations well, consecutive quarter, consumer electronics, continue drive, continue execute, continue improve$^+$, continue see, continued focus, continued growth, continued improvement$^+$, continued momentum, continued strong$^+$, continuing push, contributed improvement$^+$, cost control, cost management, couple years, credit card, curious guys, customers increasing, data centers, debt reduction, delivered outstanding, delivered strong$^+$, demand across, demand coming, different types, done great$^+$, driven better$^+$, driven improved$^+$, driven strong$^+$, driving growth, drove strong$^+$, due improved$^+$, due strong$^+$, early innings, earnings per, end markets, even better$^+$, even higher, exceeded expectations, exceeded guidance, exceeded high, exceeding expectations, exceeding guidance, excellent quarter$^+$, excellent results$^+$, exceptional quarter$^+$, excess cash, executed well, executing well, execution quarter, execution team, existing accounts, expanded basis, expanding gross, expanding operating, expansion existing, expansion over, expect continue, extremely pleased$^+$, extremely proud, fantastic job$^+$, favorable product$^+$, few years, financial performance, financial results, finish year, first congratulations, free cash, further improving$^+$, gaining share$^+$, generated cash, generated free, generated operating, getting good$^+$, getting more, given strength$^+$, given strong$^+$, gives pretty, going direction, going somewhat, good growth$^+$, good job$^+$, good momentum$^+$, good quarter$^+$, good results$^+$, good see$^+$, good start$^+$, good strength$^+$, great customer$^+$, great execution$^+$, great growth$^+$, great hear$^+$, great job$^+$, great quarter$^+$, great results$^+$, great see$^+$, great start$^+$, great way$^+$, great year$^+$, grew compared, grew sequentially, growth across, growth came, growth coming, growth driven, growth quarter, growth saw, growth seeing, guidance up, guys congratulations, halo effect, helped drive, helped quarter, helping drive, high end, higher asps, higher last, higher revenue, higher revenues, higher volumes, highest level$^+$, hitting cylinders, impressive quarter$^+$, improved basis$^+$, improved demand$^+$, improved financial$^+$, improved fourth$^+$, improved gross$^+$, improved operating$^+$, improved outlook$^+$, improved profitability$^+$, improved significantly$^+$, improvement across$^+$, improvement adjusted$^+$, improvement basis$^+$, improvement compared$^+$, improvement driven$^+$, improvement gross$^+$, improvement operating$^+$, improvement over$^+$, improvement performance$^+$, improvement primarily$^+$, improvement profitability$^+$, improvement quarter$^+$, improvement reflects$^+$, improvements gross$^+$, improvements operating$^+$, improving margins$^+$, improving operating$^+$, income continuing, income improvement$^+$, income increased, income margin, income quarter, income up, increase adjusted, increase compared, increase demand, increase fiscal, increase gross, increase guidance, increase margin, increase market, increase over, increase prior, increase product, increase revenue, increase sequentially, increased adjusted, increased basis, increased demand, increased efficiencies$^+$, increased full,

increased guidance, increased over, increased prior, increased revenue, increased sequentially, increased visibility, increasing full, increasing guidance, inflection point, initiatives around, initiatives implemented, introduced last, job quarter, just curious, just great^{+}, just speak, just strength^{+}, just strong^{+}, keep up, key driver, level profitability^{+}, leverage operating, leverage quarter, like nice, like thank, line growth, little color, look opportunities^{+}, lot hard, lot initiatives, lot leverage, lot runway, lot work, lower raw, make announcement, manufacturing efficiencies^{+}, margin expanded, margin expansion, margin improved^{+}, margin improvement^{+}, margin increased, margin operating, margin performance, margin target, margin up, margins improved^{+}, mentioned strong^{+}, mix better^{+}, momentum business, momentum first, momentum seeing, more disciplined, more efficient^{+}, more excited^{+}, more more, more opportunity^{+}, more targeted, most growth, most interested, need add, net debt, net income, new record, nice growth, nice improvement^{+}, nice job, nice leverage, nice quarter, nice results, nice see, nice thing, number go, obviously good^{+}, obviously great^{+}, obviously guys, obviously nice, obviously pleased^{+}, obviously strong^{+}, okay great^{+}, operating efficiencies^{+}, operating income, operating leverage, operating margin, operating performance, operational execution, operations increased, organic growth, outstanding performance, outstanding quarter, outstanding results, over guidance, over last, over prior, overall pleased^{+}, particularly pleased^{+}, particularly strong^{+}, per square, percentage revenue, percentage sales, performance across, performance first, performance fourth, performance really, performance result, pleased financial^{+}, pleased first^{+}, pleased fourth^{+}, pleased performance^{+}, pleased progress^{+}, pleased quarter^{+}, pleased report^{+}, pleased results^{+}, pleased second^{+}, pleased strong^{+}, point anything, point expansion, point improvement^{+}, points compared, points higher, points sequentially, positive adjusted^{+}, positive momentum^{+}, positively impacted^{+}, pretty excited^{+}, pretty good^{+}, pretty impressive^{+}, pretty much, pretty nice, pretty strong^{+}, previous year, product innovation^{+}, profit increase, profitable quarter^{+}, progress made^{+}, provided operating, quality revenue, quarter across, quarter exceeded, quarter exceeding, quarter generating, quarter great^{+}, quarter guys, quarter improved^{+}, quarter improvement^{+}, quarter increase, quarter increased, quarter nice, quarter performance, quarter record, quarter row, quarter strong^{+}, quarter up, quick questions^{+}, raise guidance, raised guidance, raising full, raising guidance, range representing, range up, reaching new, really driving, really good^{+}, really great^{+}, really happy^{+}, really helped, really helpful, really helping, really impressive^{+}, really nice, really pleased^{+}, really really, really resonating, really solid, really starting, really well, record adjusted, record company, record fourth, record high, record net, record operating, record quarter, record quarterly, record results, record revenue, record sales, record second, record up, reduction initiative, report second, report strong^{+}, represents growth, repurchase program, result better^{+}, result improved^{+}, result strong^{+}, results across, results better^{+}, results demonstrate, results driven, results exceeded, results strong^{+}, revenue exceeded, revenue grew, revenue growth, revenue improved^{+}, revenue increase, revenue increased, right direction, room go, sales improved^{+}, sales increase, sales increased, sales up, sales wholesale, saw good^{+}, saw nice, say pleased^{+}, second question^{+}, seeing benefit^{+}, seeing benefits, seeing good^{+}, seeing growth, seeing strength^{+}, sequential improvement^{+}, sequential increase, sequentially result, share above, share buyback, share count, share exceeded, share gain^{+}, share gains^{+}, share improvement^{+}, share increased, share repurchase, share up, significant improvement^{+}, significant improvements^{+}, significantly exceeded, significantly improved^{+}, special dividend, square foot, start year, starting see, still lot, still small, stock buyback, stock repurchase, strategic relationships, 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 focus^{+}, strong fourth^{+}, strong growth^{+}, strong momentum^{+}, strong operating^{+}, strong performance^{+}, strong quarter^{+}, strong quarterly^{+}, strong results^{+}, strong revenue^{+}, strong second^{+}, strong sequential^{+}, strong start^{+}, strong third^{+}, strong top^{+}, stronger anticipated^{+}, stronger expected^{+}, strongest quarter^{+}, success driving^{+}, success seeing^{+}, success strategy^{+}, summary pleased^{+}, taking share, talk little, tangible results, tax rate, team delivered, team done, team great^{+}, teams done, terrific quarter, thank congratulations, thanks comments, thanks lot, thing add, think able^{+}, think continue, think starting, think sustainable, think team, third consecutive, top bottom, tremendous momentum^{+}, try make, under facility, up basis, up compared, up over, up previous, up sequentially, upside quarter, use capital, value price, wanted ask, well above, well against^{+}, well ahead, well balanced, well great^{+}, well together, wondering might, world just, year fourth, year raising, year results, year strong^{+}, years ago, years get.

Negative bigrams: able predict^{+}, actions address, actions taking, actually couple, additional cost, address current, address issues, addressing issues, adverse impact^{+}, adversely impacted^{+}, again think, aggressive marketing, aggressive price, aggressive pricing, already discussed, also affected, also impacted, also impacting, also negatively^{+}, ancillary services, approximately decline^{+}, average monthly, back track, back up, balance year, beat dead, became clear, became more, believe better^{+}, believe changes, believe important, believe prudent, believe right, believe underlying, below expectation, below expectations, below guidance, below low, biggest impact, bit late^{+}, bit longer, bit time, bookings second, break even^{+}, bright spots, business decline^{+}, business not, came below, came up, capital raised, carrying value, cash payments, certain areas, challenges business^{+}, challenges face^{+}, challenges not^{+}, challenges quarter^{+}, challenging quarter^{+}, change estimate, change fair, change guidance, change sales, changed last, changes making, changes not, clear not, come back, coming lower, coming quarter, company files, compared earnings, compared positive^{+}, competitive activity, competitive environment, competitive pressure, competitive pressures, competitive pricing, completed quarter, confident new^{+}, corrective actions, cost overruns^{+}, costs associated, costs increased, costs incurred, couple months, current forecast, current market, currently working, customer programs, day rates, dead horse, debt covenants, decision making, decision not, decision reduce, decisions around, decline driven^{+}, decline due^{+}, decline gross^{+}, decline primarily^{+}, decline quarter^{+}, decline related^{+}, decline revenue^{+}, decline revenues^{+}, decline sales^{+}, declined quarter^{+}, decrease approximately, decrease cash, decrease net, decrease prior, decrease product, decrease revenue, decreased compared, decreased due, decreased first, dedicated sales, despite challenges^{+-}, difficult forecast^{+}, disappointed results^{+}, discount rate, discuss more, discussed earlier, discussion over, down basis, down compared, down down, down last, down low, down most, down much, down period, down points, down previous, down prior, down quarter, down relative, down second, down third, down versus, down year, driven lower, due current, due decrease, due decreased, due delays^{+}, due economic, due inventory, due lower, due primarily, due timing, earlier not, earnings down, effectively compete, ended increase, enough offset, equity financing, equity income, even though, excess inventory, exchange impacts, excluding litigation^{+}, execute better^{+}, execution issues, expect begin, expense increased, expenses related, fact not, factors impacted, factors impacting, factors led, fair question^{+}, fall short, fast enough, fell short, few days, few minutes, first few, first weeks, fiscal first, flow negative^{+}, further deterioration^{+}, generation products, get back, get hit, get profitability^{+}, get worse^{+}, gives confidence, go back, going away, going get, going take, got pushed, got work, greater expected^{+}, growing pains, guess just, guess question^{+}, guidance change, guidance down, guidance question^{+}, guidance reduction, guide down, guys thanks, half quarter, help understand, high confidence, higher cost, however believe, impact changes, impact lower, impact third, impact transition, impacted business, impacted first, impacted lower, impacted margins, impacted quarter, impacted revenue, impacting revenue, income decreased, income down, increase cost, increase expenses, increase inventory, increased competition, increased cost, increased costs, inherent risks, initially expected, inventory correction^{+}, inventory not, issue just, issue not, issue quarter, issue really, issues facing, issues mentioned, issues not, issues really, just clear, just got, just lower, just matter, just not, just seems, just trying, just want, kind negative^{+}, labor costs, lack visibility^{+}, large customer, last days, last weeks, late quarter^{+}, leadership changes^{+}, led lower, levels within, like discuss, line guidance, line prior, little confused^{+}, local advertising, long term, longer expected, look back, losing market^{+}, losing share^{+}, loss attributable^{+}, loss compared^{+}, loss continuing^{+},

loss per$^+$, loss quarter$^+$, loss revenue$^+$, loss third$^+$, lost business$^+$, lost revenue$^+$, lost sales$^+$, lost share$^+$, low end, lower anticipated, lower demand, lower end, lower expectations, lower expected, lower gross, lower guidance, lower level, lower margin, lower margins, lower net, lower pricing, lower revenue, lower revenues, lower sales, lower volume, lower volumes, macro issues, made adjustments, made decision, made number, maintain market, make change, make decisions, make numbers, make sure, make up, making adjustments, making necessary, management changes, many know, many more, margin compression, margin decline$^+$, margin declined$^+$, margin decreased, margin impact, margin pressure, margin pressures, margins decreased, margins impacted, market declined$^+$, may continue, mean guess, meet expectations, mentioned earlier, mix issue, mix issues, month period, month quarter, more aggressive, more anticipated, more cautious, more challenging$^+$, more competitive, more conservative, more expected, more gradual, more pressure, more significant, more time, most notably, most significant, moving faster, moving pieces, much lower, near term, need address, need improve$^+$, need now, negative cash$^+$, negative free$^+$, negative impact$^+$, negatively impacted$^+$, net loss$^+$, new bookings, new guidance, new leadership$^+$, new marketing, new reality, normally expect, not able$^+$, not acceptable, not anticipated, not come, not deliver, not effective$^+$, not enough, not execute, not expected, not getting, not going, not good$^+$, not happen, not happy$^+$, not hit, not issue, not losing$^+$, not lost$^+$, not make, not materialize, not meet, not offset, not perform, not performing, not pleased$^+$, not related, not satisfied$^+$, not seeing, not think, now saying, number issues, occurred quarter, offset improved$^+$, offset lower, offset partially, okay final, okay just, okay thank, operating environment, operational challenges$^+$, optimize performance, orders come, overall market, part challenge$^+$, part issue, part quarter, partial quarter, particularly weak$^+$, patient care, paying close, payments due, per quarter, perfect storm$^+$, performance issues, period time, pick up, plan launch, plans place, point reference, positive note$^+$, pressure gross, pressure seeing, previously anticipated, previously expected, price competition, price declines$^+$, pricing issue, pricing pressure, primarily impacted, primarily lower, primarily related, prior period, products want, projects delayed$^+$, promotional activity, providing guidance, pull back, push outs, pushed back, putting pressure, qualification process, quarter back, quarter below, quarter challenging$^+$, quarter decrease, quarter decreased, quarter down, quarter impact, quarter impacted, quarter loss$^+$, quarter next, quarter not, quarter primarily, quarter progressed$^+$, quarter second, ramping up, reduce cost, reduce operating, reduced demand, reduction gross, reduction guidance, reduction sales, related acquisition, related implementation, remain confident$^+$, reported loss$^+$, reported period, result lower, result not, resulted lower, resulting decline$^+$, results below, results not, return growth, revenue adjustment, revenue decline$^+$, revenue declined$^+$, revenue declines$^+$, revenue decrease, revenue down, revenue expectations, revenue outlook, revenue shortfall$^+$, revenue trend, revenues declined$^+$, revenues decreased, revenues down, revenues net, revised expectations, revised full, right decision, right strategy, right thing, said not, sales accounted, sales cycle, sales declined$^+$, sales decreased, sales execution, sales fell, sales infrastructure, saw lower, saw release, saw slowdown$^+$, saw weakness$^+$, saying going, scale back, second issue, see weakness$^+$, seen decline$^+$, seen more, segment decreased, senior executives, senior living, settle down, several factors, several large, share down, share loss$^+$, ship product, short expectations, short term, show up, significant decline$^+$, significant portion, significantly lower, slightly below, slow down$^+$, slow start$^+$, slowdown business$^+$, slowed down$^+$, slower anticipated$^+$, slower expected$^+$, slower growth$^+$, slower ramp$^+$, slower sales$^+$, slower start$^+$, slowing down$^+$, slowing growth$^+$, soft demand, something else, somewhere around,

sounded like, sounds like, specific issues, specifically related, start buying, step back, still believe, still profitable^{+}, stock down, strategic review, sure understand, take longer, take time, taken longer, takes time, taking actions, taking aggressive, taking down, taking longer, taking more, taking steps, tax asset, team place, teams place, term believe, term loans, terms sequential, things down, think issue, think takes, third fourth, though say, thought going, timing issue, timing issues, timing revenue, timing sales, took down, took longer, towards lower, traffic trends, transactional business, transition company, transition new, transition period, try understand, trying figure, trying get, trying reconcile, trying understand, under pressure, under way, unusually high, up running, used operations, volumes down, wages benefits, want make, wanted clarify, weaker expected^{+}, weaker sales^{+}, weakness saw^{+}, weeks quarter, well understand, within guidance, within next, worse expected^{+}, year based, year down, year not.